NASA TECHNICAL TRANSLATION NASA TT F-15,460

SIGNAL HARMONIC ANALYSIS USING THE HP 2116 B COMPUTER

F. Palutan

(NASA-TT-F-15460) SIGNAL HARMONIC ANALYSIS USING THE HP 2116 B COMPUTER (Kanner (Leo) Associates)

N74-19839

CSCL 09B

Unclas
G3/08 34394

Translation of "Sistema per l'analisi armonica dei signali con il calcolatore HP 2116 B," National Research Council, Rome, Italy, Research and Technology Laboratory for the Study of Plasma in Space, LPS-72-17, July 1972 (STAR N74-12905), 74 pp.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 APRIL 1974

STANDARD TITLE PAGE

1. Report No. NASA TT F-15,460	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle SIGNAL HARMONIC ANALYSIS USING THE HP 2116 B COMPUTER		5. Report Date April 1974
		6. Performing Organization Code
7. Author(s) F. Palutan, Research and Technology Laboratory for the Study of Plasma in Space		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Leo Kanner Associates, P.O. Box 5187, Redwood City, California 94063		11. Contract or Grant No. NASw-2481
		13. Type of Report and Period Covered Translation
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, WASHINGTON, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Translation of "Sistema per l'analisi armonica dei segnali con il calcolatore HP 2116 B," National Research Council, Rome, Italy, Research and Technology Laboratory for the Study of Plasma in Space, LPS-72-17, July 1972 (STAR N74-12905), 74 pp.

16. Abstract Computer programs for real-time signal harmonic analysis as applied to the HP 2116 B computer are described. The BASIC language is used throughout. Topics discussed include sampling, quantization, discrete and fast Fourier transform algorithms, truncation, and amplitude spectrum recovery by Fourier transform of the autocorrelation function. The flow charts of the main program and subroutines are presented and technical details of the programs outlined. Experimental results are detailed for the case of signals embedded in noise with various S/N ratios. The effects of aliasing and the use of Hanning windows are discussed. Extensions both in hardware and software of the present system are reviewed.

17. Key Words (Selected by Author(s))	18. Distribution Statement Unclassified - Unlimited
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified

NASA-HQ

Table of Contents

SIGNAL HARMONIC ANALYSIS USING THE HP 2116 B COMPUTER

F. Palutan
Research and Technology Laboratory for the Study of Plasma in Space

1. Introduction /1*

Two procedures are generally used to process data coming from a physical experiment or from some other source: the first, deferred in time, consists in storing the data in a suitable fashion and analyzing them at a second time, making use of rough systems of calculation, if necessary.

The second procedure, in real time, allows the data to be processed at the very moment at which they are produced; this method allows the experimenter to observe the evolution of the phenomenon directly, and to intervene in the instrumentation at his disposal in such a manner as to obtain the maximum quantity of information.

The techniques of analysis in real time are used in many fields, from physical measurements to medical applications [1]; the present work originated as a feasibility study of a system for the analysis of data coming from an infrared telescope, but the conclusions are general, and they can be applied to different problems.

The function of the analysis time in many applications is composed of a deterministic, or signal component, and of a casual, or noise component.

In these cases, the first problem of the analysis is to separate the signal from the noise, which can be done with different methods, according to the type of signal and the information /2
that is desired.

The second problem is that of studying the properties of the signal, such as, for example, flow in the time domain or in that of frequency. The power spectrum constitutes one of the most important and most useful instruments of analysis for obtaining the properties of a signal.

In the case of deterministic signals of known form, although the flow in the frequency domain does not add information to that obtainable by an analysis in the time domain, it is useful to indicate the most important characteristics of the signal; however, if there is no preliminary information available, e.g. because the signal is embedded in the noise, then spectral analysis is

*Numbers in the margin indicate pagination in the foreign text.

indispensable for obtaining the most significant parameters. Only when the frequency of the signal is known a priori can different spectral analysis techniques be used to separate the signal from the noise, such as, for example, the use of tuned filters.

In the case of casual signals, such as the noise produced by an electronic device, the power spectrum is of fundamental importance, since it constitutes the only investigational means of analyzing the statistical properties of such phenomena.

In the present work, the principle objective is considered to be detection of a deterministic signal embedded in noise, by means of determination of the power spectrum.

The techniques generally used to determine the power spectrum of a time function are represented in Fig. 1 [2, 3, 4]. /3

In the first diagram (Fig. 1a), use is made of analog instrumentation; the integrator output supplies the amplitude of the spectrum in the frequency interval corresponding to the passband of the filter. To obtain the spectrum of a signal, it is necessary that the filter passband flow between two assigned frequency limits with a flow speed compatible with the integrator time constant; the spectrum resolution is the ratio of the frequency interval swept and the filter passband.

Figure 1b shows the diagram for data processing used in the present work; the analog signal, converted into numerical form, is acquired from a computer that works out the Fourier transform and the power spectrum of the sample sequence representing the signal, according to the algorithms discussed below.

In the last diagram, the power spectrum is calculated as a Fourier transform of the autocorrelation function, according to the Wiener-Khinchin theorem.

Variance in the results is diminished by means of an averaging operation, carried out by the integrator in the first diagram and by the computer in the other two.

The choice of one of the three methods depends on the particular problem that must be resolved; the analog systems are more economical than the digital ones and are the only ones available in the area of high frequencies. /4

On the other hand, diffusion of small computers has quite often made the digital processing of signals advantageous, due to the accuracy with which the results are calculated [5, 6].

The choice between the two digital systems of analysis (Fig. 1b and 1c) depends on the problem being examined; for example, in the case of an infrared measurement, the interferometers commonly used supply an output signal that is the autocorrelation of the input signal, for which the only method of obtaining the power spectrum is that outlined in Fig. 1c, in which the first two blocks, which compute the autocorrelation function, are substituted by an interferometer.

In general, however, the computation scheme of Fig. 1b is used, since it is faster than that of Fig. 1c, inasmuch as it is not necessary to compute the autocorrelation function.

In every case, the use of these computation schemes together is often possible with the same Fourier analyzer, simply by calling different programs of calculation.

In the present study, the performance and characteristics of a system based on the HP 2116 B computer, which operates according to the plan in Fig. 1b, are analyzed.

2. Signal Harmonic Analysis with Digital Techniques /5

The procedures for calculating the spectrum of a signal by means of the scheme in Fig. 1b will be discussed in detail in this section. We will examine successively the various processing steps undergone by the signal, assumed to be ergodic and stationary, from sampling up to the averaging operation.

2.1. Sampling

Sampling consists in considering the values assumed by the signal to be continuous in time at determined instants, generally at constant time intervals (Fig. 2).

The following value is defined as the n-th sample:

$$x_n = x(t_o + n\,\Delta t) \qquad (2.1)$$

where t_o is the initial instant and Δt is the sampling interval.

The sampling theorem [7] states that the interval Δt is linked to band B of the signal by the relation:

$$\Delta t \leq \frac{1}{2B} \qquad (2.2)$$

The value f_N which satisfies the relation: /6

$$f_N = \frac{1}{2\Delta t} \tag{2.3}$$

is called the "Nyquist frequency."

For a correct sampling, it is necessary that the signal band B be less than the Nyquist frequency f_N; this can be obtained with a suitable choice of the sampling interval Δt.

If relation (2.2) is not verified, the phenomenon of aliasing takes place; this is illustrated qualitatively in Fig. 3a.

The very high frequency sinusoid, sampled at time intervals Δt that are too large, is not distinguishable in any way after sampling from the very low frequency sinusoid. It is shown that all the lines at frequency $f_N \leq f < 2f_N$ are represented by lines at frequency $2f_N - f$; the spectrum is continuous, and it is inverted specularly around the frequency f_N (Fig. 3b). The inversion process continues also for frequencies higher than $2f_N$ around the integral multples of f_N.

Once Δt is fixed, on the basis of the signal band and the characteristics of the sampling circuit, it is appropriate to filter the signal with a low-pass filter with cut-off frequency less than or equal to f_N, so as to eliminate possible components of frequency /7
$f > f_N$, which could alter the spectrum flow because of aliasing.

2.2. Quantization

The signal samples, defined by (2.1), can assume all the values included in the continuous interval in which function x(t) is defined. The quantization operation consists in assigning the mean value x_k to all the x(t) values included in a certain predetermined interval:

$$x_k - \frac{\Delta x}{2} \leq x < x_k + \frac{\Delta x}{2} \tag{2.4}$$

The amplitude of the quantum Δx depends on the interval X of definition of x(t), and on the number of bits b available for representing the function in numerical form. In fact, if a different value is assigned to every quantization interval, with b bits, 2^b intervals can be enumerated for which Δx is given by:

$$\Delta x = \frac{X}{2^b} \tag{2.5}$$

Converting the signal into numerical form introduces a quantization error, given by:

$$\epsilon = x - \bar{x}_k \qquad (2.6)$$ /8

In the hypothesis that in the quantization interval (2.4) function x assumes all values with equal probability, then the mean square error due to quantization is given by:

$$\bar{\epsilon}^2 = \frac{1}{\Delta x}\int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} \epsilon^2 \, d\epsilon = \frac{\Delta x^2}{12} \qquad (2.7)$$

For a large enough number of bits b, this error has a very small value as compared to other causes of error introduced in signal harmonic analysis, chiefly as compared to the noise that is generally superposed on the input signal. In our case, in which the conversion takes place with 9 bits, the quantization error will not be considered, since it is negligible in comparison with the other errors discussed below.

2.3. Discrete Fourier Transform and Power Spectrum

For a continuous signal, the Fourier transform is expressed by the integral:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt \qquad (2.8)$$

which, in the case of signals limited in time in the interval (0,T), /9
becomes:

$$X(f,T) = \int_{0}^{T} x(t)\, e^{-j2\pi ft}\, dt \qquad (2.9)$$

When the signal is sampled, a discrete Fourier transform (DFT) can be defined, in a manner analogous to (2.9):

$$X(f_k, T) = \Delta t \sum_{n=0}^{N-1} x(n \Delta t) e^{-j2\pi f n \Delta t} \tag{2.10}$$

with N Δt = T.

One speaks of discrete frequencies such as those defined according to the relation:

$$f_k = \frac{k}{T} = \frac{k}{N \Delta t} \tag{2.11}$$

with k = 0, 1, 2, ..., N - 1.

On the basis of (2.11), (2.10) becomes:

$$X_k = \frac{X(f_k, T)}{\Delta t} = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N} kn} \tag{2.12}$$

where the first term, generally complex, has been divided by Δt for normalization.

From relations (2.11) and (2.3) it is found that the Nyquist frequency corresponds to k = N/2; only the Fourier coefficients x_k with k < N/2 are independent, while for k $\geq$ N/2, they are repeated identically. /10

The Fourier coefficients can be used to calculate N/2 lines of the power spectrum at distance 1/T, by the relation:

$$G_k = \frac{2\Delta t}{N} |X_k|^2 \tag{2.13}$$

where $|X_k|^2$ is the square of the modulus of X_k.

2.4. Truncation Effect

A physical phenomenon, described by a function of time $x(t)$ can be known only in a finite time interval; based on this consideration, the transform $x(f,T)$ is defined by relation (2.9). It is important to examine the consequences of truncation of $x(t)$ in the calculation of the power spectra.

A function $x(t)$ defined in the limited interval $(-T/2, T/2)$ can be considered the product of a function $y(t)$, defined in the interval $(-\infty, \infty)$, such that $x(t) = y(t)$ in the interval $(-T/2, T/2)$, and of the function $u(t)$ such that:

$$u(t) = 1 \text{ for } -T/2 \leq t \leq T/2 \qquad = 0 \text{ for } |t| > T/2 \tag{2.14}$$

The function $u(t)$ is called the rectangular window. /11

The Fourier transform of the function $x(t)$ can then be defined in the following manner:

$$X(f,T) = \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{-j2\pi ft}\,dt = \int_{-\infty}^{\infty} y(t)u(t)e^{-j2\pi ft}\,dt \tag{2.15}$$

On the basis of a known harmonic analysis theorem which establishes the fact that the transform of a product of two functions is given by the convolution integral of the two transforms, the following is obtained:

$$X(f,T) = \int_{-\infty}^{\infty} Y(f')\,U(f-f')\,df' \tag{2.16}$$

where $Y(f)$ and $U(f)$ are the transforms of the two functions $y(t)$ and $u(t)$, and the second term of (2.16) is the convolution integral of $Y(f)$ and $U(f)$.

Let us suppose that the function $x(t)$ is a truncated sinusoid, of frequency f_0; the transforms of $y(t)$ and $u(t)$ are, respectively:

$$Y(f) = \delta(f-f_o)$$
$$U(f) = T\left(\frac{\sin \pi f T}{\pi f T}\right) \tag{2.17}$$

By applying (2.16), the following is obtained: /12

$$X(f,T) = U(f-f_o) \tag{2.18}$$

which is represented by the solid curve in Fig. 4, translated by a quantity f_o on the axis of the abscissas.

It can be stated that the truncation effect is thus a resolution loss in calculation of the spectrum: in the previous example, the truncated sinusoid is not represented by a pulse function, but by a function of the type sin x/x.

In the case of calculating a spectrum beginning with a time series of N numbers, it is possible to obtain, as has been seen, N/2 estimates of the spectrum; the frequency resolution is thus given by:

$$\Delta f = \frac{2 f_N}{N} = \frac{1}{\Delta t \cdot N} = 1/T \tag{2.19}$$

An effect that is linked to the choice of the window's rectangular shape u(t) is the presence of leading and trailing edges with amplitude equal to about 20% of that of the top. Consequently, the spectrum is distorted, and this can make the intepretation of the data difficult. There exist windows in which the leading and trailing edges are damped.

One of the most common is that termed the "Hanning," defined by the expression:

/13

$$\begin{aligned} v(t) &= \tfrac{1}{2}\left(1+\cos\frac{2\pi t}{T}\right) && \text{for } |t| \leq \frac{T}{2} \\ &= 0 && \text{for } |t| > \frac{T}{2} \end{aligned} \tag{2.20}$$

The flow of this function in the frequency domain is given by the dashed curve in Fig. 4; the Hanning window offers a wider t o p than the rectangular window, and thus less resolution and better damping of the leading and trailing edges; the resultant spectrum is therefore less distorted.

2.5. Variance in the Spectral Estimates

A variance is associated with each coefficient of the power spectrum calculated by (2.13); this variance will be calculated in this paragraph in order to establish the reliability of the results obtained with the calculation process discussed up to this point.

From (2.12) and (2.13), it is found that the k-th coefficient of the power spectrum of a function x(t), defined in the interval (0,T), is given by:

$$G(f) = \frac{2}{T} |X(f,T)|^2 \tag{2.21}$$

where:

$$X(f,T) = \int_0^T x(t)\, e^{-j2\pi ft}\, dt \tag{2.22}$$

The spectral coefficients are calculated for discrete values /14
of frequency f, at distance $\Delta f = 1/T$.

The Fourier transform X(f,T) is a linear operation in the complex field, for which, if function x(t) has a normal distribution, the real and imaginary components of X(f,T), which are $X_R(f,T)$ and $X_I(f,T)$, respectively, also have an independent normal distribution. It results that the quantity:

$$|X(f,T)|^2 = X_R^2(f,T)+X_I^2(f,T) \tag{2.23}$$

has a known distribution with the name "χ_n^2 distribution," where n is the number of degrees of freedom corresponding to the number of independent variables in the second term of (2.23). The average and the variance of a distribution χ_n^2 are, respectively:

$$E[\chi_n^2] = \mu_{\chi_n^2} = n$$
$$E[(\chi_n^2-\mu_{\chi_n^2})^2] = \sigma_{\chi^2}^2 = 2n \tag{2.24}$$

It is then found that the relative standard error of each estimate of the power spectrum is given by:

$$\varepsilon_r = \frac{\sigma_{\chi^2}}{\mu_{\chi^2}} = \frac{\sqrt{2n}}{n} = \sqrt{\frac{2}{n}} \qquad (2.25)$$

Substituting n = 2 in (2.25), $\varepsilon_r = 1$ is obtained; this means that the standard deviation of an estimate is equal to the expected value of the estimate itself, which is unacceptable in the greater part of the applications. /15

In addition, it results from (2.25) that the standard deviation is independent of the length of the record; this means that an increase in the number of samples increases the spectral resolution, i.e. the number of estimates for the same signal band; but it does not reduce the standard error.

In order to reduce the error in calculating the spectrum, two equivalent paths may be followed: complex spectrum averaging can be carried out, or averaging of contiguous spectral estimates can be done.

In the first case, it is a matter of averaging a certain number K of spectral estimates, each obtained from signals observed for time T_k; the total length of observation time (record length) is thus: $T = KT_k$, and the final spectral resolution is: $\Delta f = 1/T_k$.

It is also possible to proceed by first processing the signal acquired during time interval T, obtaining a resolution $\Delta f' = 1/T$; averaging of K contiguous estimates of the spectrum is then carried out, and resolution $\Delta f = K\Delta f' = 1/T_k$ is thus obtained.

In both cases the value of n to be substituted in (2.25) is $n = 2k = 2T\cdot\Delta f$, for which the relative standard error becomes:

/16

$$\varepsilon_r = \frac{1}{\sqrt{K}} = \frac{1}{\sqrt{\Delta f T}} \qquad (2.26)$$

From the practical point of view, the first process, which involves the execution of complex averaging, is preferable to the second, since it allows the calculations to be carried out on a constant number of samples, whatever the value of K, and thus to occupy a fixed, limited part of the computer memory in the data acquisition phase.

3. Description of the System for Calculating the Amplitude Spectrum /17

In this chapter the realization of the system for processing signals in real time will be discussed in detail.

After the description of the algorithm of the fast Fourier transform (FFT), the block diagram of the system (hardware) will be discussed, and then the programs (software) will be analyzed.

3.1. Fast Fourier Transform (FFT) [8, 9, 10]

This term indicates an algorithm for calculating the discrete Fourier transform (2.12), which is carried out by a computer in a period of time that is considerably reduced compared to traditional algorithms. The introduction of the FFT in 1965 by Cooley and Tukey permitted the construction of systems for data acquisition and analysis in real time with digital processing of small dimensions.

The Fourier transform defined by (2.12) can be written:

$$X_k = \sum_{n=0}^{N-1} x_n W^{kn} \qquad k = 0, \ldots, N-1 \tag{3.1}$$

where: /18

$$W = e^{-j\frac{2\pi}{N}}$$

Two methods exist for the fast calculation of the N Fourier coefficients expressed in (3.1), called time decimation or frequency decimation, respectively. They are equivalent from the standpoint of calculation time; here only the second will be described, since that is the method used in the present work.

To simplify the calculations, N generally is considered as an integral power of 2, which does not limit the use of the algorithm.

Let the sequence of N numbers x_n be divided into two equal parts, each of N/2 points, formed from the first half and the second half of the original series, respectively; the following is obtained:

$$y_n = x_n$$
$$z_n = x_{n+N/2} \tag{3.2}$$
$$n = 0, 1, \ldots, (N/2-1)$$

(3.1) can then be written in the form: /19

$$X_k = \sum_{n=0}^{N/2-1} \left[y_n \exp(-j2\pi kn/N) + z_n \exp(-j2\pi k(n+N/2)/N)\right]$$

that is:

$$X_k = \sum_{n=0}^{N/2-1} \left[y_n + z_n \exp(-j\pi k)\right] \exp(-j2\pi kn/N) \tag{3.3}$$

It is convenient to consider the coefficients with even index (k = 2h) and those with odd index (k = 2h + 1) separately.

In the first case, Eq. (3.3) yields:

$$\begin{aligned} R_h = X_{2h} &= \sum_{n=0}^{N/2-1} \left[y_n + z_n\right] \exp\left[(-j2\pi hn)/(N/2)\right] \\ &= \sum_{n=0}^{N/2-1} \left[y_n + z_n\right] W^{2hn} \end{aligned} \tag{3.4}$$

with $0 \leq h < N/2$.

This is the expression of the Fourier coefficients in the series $(y_n + z_n)$ of N/2 points.

For an odd k (k = 2h + 1), (3.3) becomes:

$$\begin{aligned} S_h = X_{2h+1} &= \sum_{n=0}^{N/2-1} \left[y_n + z_n \exp(-j\pi(2h+1))\right] \exp\left[-j2\pi(2h+1)n/N\right] \\ &= \sum_{n=0}^{N/2-1} \left[y_n - z_n\right] \exp\left[-j2\pi n/N\right] \cdot \exp\left[(-j2\pi hn)/(N/2)\right] = \\ &= \sum_{n=0}^{N/2-1} \left[\ldots \right. \end{aligned} \tag{3.5}$$

with $0 \leq h < N/2$. /20

This relation yields the Fourier coefficients of the series of N/2 points:

$$(y_n - z_n)W^n$$

It can be concluded that the N Fourier coefficients of the primary sequence x_n can be calculated by means of the transforms of two series of N/2 numbers, each of which is a linear combination of two numbers of the original series.

Figure 5 illustrates the graph of the operations for N = 8.

For $N = 2^{\ell}$, this process can be repeated ℓ times, until the number of points on which the first transform is operated is reduced to 2.

The signal flow chart is reported in Fig. 6 for N = 8.

The reduction in calculation time obtained by following this plan is determined approximately by taking into account only the multiplications, and neglecting the additions and control operations. Calculation of a coefficient X_r with (3.1) involves the execution of N multiplications in the complex field; for the integral transform, the number of multiplications is N^2. In the graph in Fig. 6, the number of multiplications for every step in the complex field is N, the number of steps is $\log_2 N$, and thus the total number of multi- /21
plications is $N\log_2 N$.

The reduction in calculation time is about 98% for N = 512, and 99% for N = 1024.

In the graph in Fig. 6, the Fourier coefficients obtained at the end of the process can be ordered; the order of the indices is reversed (bit-reversed), that is to say, in the i-th position a coefficient with index k is found, which is obtained from the binary number "i" read in the opposite direction. This property makes the reordering of coefficients easy. An advantage of this plan is that at every step the results of processing two points can be stored in the same memory locations that contain the original points no longer used in the processing; in this manner, a single data vector can contain both the intermediate steps and the final results of the calculation.

Different data flow charts and a more complete discussion of the FFT algorithm are reported in the works cited.

3.2. Block Diagram of the System

The system for data acquisition and processing in real time is represented in Fig. 7. The HP 2116 B computer, equipped with a memory of 16,000 words of 16 bits, receives data in numerical form from the analog-to-digital converter and analyzes them, showing the /22
results on the teletype, on the graphic panel, or the video terminal.

The signal, produced by a function generator, is added to the gaussian noise coming from a noise generator. The addition signal is next filtered to eliminate frequency components larger than the Nyquist frequency, and is then converted to numerical form by the analog-to-digital converter.

The conversion command, supplied by a pulse generator, allows the computer to acquire 100 samples per second, each of 8 bits plus one bit for the sign.

After acquiring a temporal sequence of N samples, the computer carries out processing and shows the results in numerical form on the teletype, or in graphic form on the video terminal or the graphic panel.

The procedure adopted does not correspond to that commonly called "in real time," in which the acquisition of new data takes place contemporaneously with processing of the numerical sequence acquired before; in the system in Fig. 7, the two phases -- data acquisition and processing -- are consecutive. This procedure is simpler from the standpoint of programming, and it does not restrict the results obtained if the hypothesis is made that the input signals are stationary, that is, the statistical parameters which characterize the signals themselves do not undergo variations in time.

On the other hand, if a fast converter is not available, the /23
acquisition time is greater than the processing time; thus it does not appear to be necessary to modify the system, because it would operate in real time.

3.3. Structure of the Program

The main system control program, written in BASIC language, for all the data acquisition and processing operations calls some subroutines in ASSEMBLER language which operate on whole numbers, in order to decrease the calculation times.

The flow chart of the main program is shown in Fig. 8.

During the initializing phase, the number N of samples to be processed is supplied to the program, and then, once the sampling period is fixed, the time T of observation of the signal (record length); in addition, the values of the function

$$\cos\left(\frac{2\pi}{N}\, n\right), \quad \text{per} \ [illegible]$$

are computed and stored, with which it is possible to construct function W^{kn}, defined by (3.1), for any value of k and n.

Another initial parameter of the program is the number of spectra that must be averaged to improve the statistical reliability of the results, as was seen in Par. 2.5.

It is found from (2.26) that the variance in result is, in fact, inversely proportional to the root of the number of averages; and in particular, an increase in the number of processes carried out should correspond to a decrease in the signal/noise ratio. /24

After initializing, the system can acquire the first sequence of N samples.

The Fourier transform of each sequence is first computed with the FFT subroutine, and the amplitude spectrum, with the POWER subroutine; the AVERAGE subroutine carries out averaging of this spectrum with the others computed before.

Data output can appear on the teletype, the video terminal, or the graphic panel; if the analysis is not finished, the program returns to the data acquisition phase.

3.4. The FFT Subroutine

The FFT subroutine computes the Fourier transform of a temporal series of data using the algorithm described in Par. 3.1, following the graph in Fig. 6.

It is deduced from this figure that the fundamental iterative operation for calculating the FFT is substitution for a pair of complex values X_m and X_n a new pair, defined as:

$$X_{m+1} = X_m + X_n \qquad (3.6)$$

$$X_{n+1} = (X_m = X_n)W^k \tag{3.6}$$ /25

where W^k is the k-th value of the N-th root of unity (3.1).

Let us consider how the moduli of the numbers vary after operation (3.6); a first indication is given by the expression:

$$\left[\frac{|X_{m+1}|^2 + |X_{n+1}|^2}{2}\right]^{\frac{1}{2}} = \sqrt{2}\left[\frac{|X_m|^2 + |X_n|^2}{2}\right]^{\frac{1}{2}} \tag{3.7}$$

which indicates that the mean square value of the moduli increases by a factor of $\sqrt{2}$ after each operation. On the other hand, from (3.6) it is found that for $|X_m|_{max} = |X_n|_{max}$ the maximum values of the moduli of the numbers in the first term of (3.7) are:

$$|X_{m+1}|_{max} = |X_{n+1}|_{max} = 2|X_m|_{max} \tag{3.8}$$

The preceding considerations are very important. In fact, all data processing is carried out with whole numbers which have a dynamic of 15 bits plus the sign; it is thus necessary to check that the results of all the arithmetic operations will not surpass the capacity of the computer's registers, i.e., in other words, that an "overflow" situation is not set up which would completely falsify the processing results.

This check can be made in the following manner: /26

-- dividing the result of each iterative operation by 2 (3.6); in this way, as can be seen from (3.8), an overflow can no longer take place.

-- checking that in the course of processing $|X_m| < 1/2\ C$ is always the case, where C is the maximum capacity of the arithmetic register (in our case, $C = 2^{15}$); if this takes place, the calculation of X_{m+1} does not produce overflow; on the contrary, all the terms intervening in the calculation of X_{m+1} are divided by 2 before initiating the arithmetic operations.

-- proceeding as in the previous paragraph, but with $|X_m| < C$; if an overflow takes place during calculation of X_{m+1}, the results of the processing already done and to be done in the course of the iteration must be divided by 2.

The first method is the most simple but the least accurate; the second was adopted in the FFT program, since it allows a rather simple check of the overflow, with loss of only 1 bit of information in the final result. The third permits utilization of the entire capacity of the registers, but it is the most complex to program.

If it is borne in mind that the data are acquired with an accuracy of 8 bits plus the sign, that the number of iterations is nine, corresponding to $N = 512$, and that the arithmetic registers of the computer have 15 bits plus the sign, an overflow can be checked in the last two iterations; the checks carried out for the program proved to be suited to yield the correct results /27
by appropriate scaling of the results.

The error introduced by this overflow checking operation was calculated, and it is seen that other causes of error, such as those discussed in Par. 2.5, weigh largely on the final result [11].

3.5. Amplitude Spectrum

With the FFT subroutine, the Fourier coefficients X_i of a temporal series of N numbers are obtained. Representation of the real and imaginary components of these coefficients supplies all the information obtainable with the harmonic analysis of a signal. At any rate, representations are usually used that make the interpretation of the data simpler, such as the power spectrum and the phase spectrum, the Nyquist and Bode graphs, etc.

In our case, we chose the representation of the amplitude spectrum, which permits immediate visualization of the frequency composition of the signal, even if it does not contain phase information.

The single power spectrum coefficient is defined by (2.21) and (2.23):

$$G_i = \frac{2}{T} \left[X_{Ri}^2 + X_{Ii}^2 \right] \tag{3.9}$$

The POWER subroutine computes the coefficients of the amplitude spectrum, defined as:

$$|X_i| = \left[X_{Ri}^2 + X_{Ii}^2 \right]^{\frac{1}{2}} \tag{3.10}$$

in such a way that the components of the small signal can be better visualized with respect to the maximum; a very simple relationship exists between the two spectra, which can be found from (3.9) and (3.10).

The root extraction operation on the second term of (3.10) is carried out by the POWER program in an approximate manner, so as to reduce the calculation time [12].

The approximate root of a number N can be calculated by making use of the approximate binary logarithm of N, in which linear interpolation between two consecutive values of the characteristic is substituted for the true mantissa (Fig. 10).

Given a number N, written in the form:

$$N = 2^{k}(1 + x) \tag{3.11}$$

with $0 \leq x < 1$, the approximate binary logarithm is defined by the expression:

$$[\log_2 N]^{*} = k + x \tag{3.12}$$

The approximate root of N is found by dividing the approximate logarithm by 2:

/29

$$[\log_2 \sqrt{N}]^{*} = \frac{k}{2} + \frac{x}{2} \tag{3.13}$$

and inversely applying the approximation used to find (3.12):

$$[\sqrt{N}]^{*} = 2^{k/2} \cdot (1 + \frac{x}{2}) \tag{3.14}$$

Calculation of (3.14) requires few simple logical operations on the computer, as compared to calculation of the root of N with the traditional algorithms.

The relative error introduced by (3.14) is given by:

$$\varepsilon = \frac{\sqrt{N} - [\sqrt{N}]^{*}}{\sqrt{N}} = \frac{(1+x)^{\frac{1}{2}} - (1 + \frac{x}{2})}{(1+x)^{\frac{1}{2}}} \tag{3.15}$$

This error is cancelled for x = 0, and it is always negative in the other cases; in addition, it increases in modulus for x tending toward 1. The maximum is obtained by substituting x = 1 in (3.15), and is equal to about 6%.

In practice, for x distributed uniformly in the interval 0 to 1, an indication of the mean error is obtained by substituting x = 0.5 in (3.15); a relative average error of about 2% is obtained.

3.6. Averaging

The AVERAGE subroutine carries out the averaging operation on all the amplitude spectra, calculated according to (3.10). The /30 averaging operation discussed in Par. 2.4 is carried out according to the relation:

$$|X_i|_J = \frac{|X_i|_{J-1} * (J-1) + |X_i|}{J} \quad (3.16)$$

where the indices J and (J - 1) represent the number of averagings carried out on the i-th component, that is, the number of acquisition and calculation cycles completed by the main program (Fig. 8).

3.7. Memory Capacity

The computer memory, which has a capacity of 16,000 words of 16 bits, is subdivided in the following manner:

Basic compiler	5000 words
Input/output system	2500 words
FFT subroutines	800 words
Data matrix	1500 words
Video matrix	500 words
Free memory	5700 words

The length of the data matrix depends on the number of samples that are processed in every acquisition and calculation cycle; it is equal to 3N, and in the table above, therefore, it is calculated for N = 500. The dimensions of the data and video matrices are specified in the main program, and they can thus be changed with great ease; the FFT programs operate with any value of N whatsoever, provided that it is an integral power of 2. /31

The control program is different according to the processing that one wishes to carry out on the data, and thus its length can also vary within the limits of the free memory zone.

4. Experimental Results /32

In this paragraph, the results of the tests carried out on the system in Fig. 7 are discussed; this simulates a signal source at different levels of the signal/noise ratio. The results are shown in the form of output on the graphic panel and printed on the teletype. The video terminal was used chiefly in the phase of setting up some programs, such as fast data output.

The signals used for the tests are the square wave and the rectangular pulse. The theoretically calculated amplitude spectra of these two signals are shown in Fig. 11. The spectrum of the square wave is made up of lines corresponding to the odd harmonics of the fundamental, with the amplitude in relation to the fundamental given by:

$$A_n/A_o = 1/n \tag{4.1}$$

The amplitude spectrum of the rectangular pulse, of duration T/8, contains all the harmonics, with amplitude variable in accordance with Fig. 11b.

Figure 12 shows the amplitude spectra of a square wave with a fundamental frequency of 5 Hz, sampled with a period of 10 msec, and filtered at the Nyquist frequency (50 Hz).

The number of samples processed is 512, and the record length is 5 sec; resolution turns out to be about 0.2 Hz. /33

The ordinate scale is calculated in the following manner. One inch corresponds to an amplitude of 7000; since the analog-to-digital converter has an accuracy of 8 bits plus the sign, and a base scale of ±1 V, there corresponds to each inch a value of the spectral component (in V/√Hz) equal to:

$$\left[\left(\frac{7000}{256}\right)^2 \cdot \frac{2\Delta t}{N}\right]^{\frac{1}{2}} = \left[(27.3)^2 \cdot \frac{2 \cdot 10^{-2}}{512}\right]^{\frac{1}{2}} = 0.17\ V/\sqrt{Hz} \tag{4.2}$$

This value can be multiplied by 2 or 4 if there was an overflow in the FFT calculation, with consequent reduction in scale.

Figures 12a and 12b show the filtered square wave and the spectrum without noise. The presence of the 11th and 13th harmonics (55 and 65 Hz) is noted; they are spread out because of aliasing; in fact, the low-pass filter whose cut-off frequency is equal to the Nyquist frequency does not completely eliminate the harmonics nearest the cut-off frequency.

In Fig. 12c, the same square wave is embedded in the noise, with a signal/noise amplitude ratio equal to 0.6. In the single spectrum (Fig. 12d), only the first and third harmonic are recognizable; after ten averagings, which reduce the variance by a factor of 3, the fifth harmonic is also visible, with a signal/ /noise ratio equal to about 0.1. /34

The same series of measurements was repeated, putting the low-pass filter at the cut-off frequency of 25 Hz, so as to eliminate the aliasing effect. As a consequence, the noise spectrum is also modified (in Figs. 13d and 13e). With a signal/noise amplitude ratio equal to 0.9, the fifth harmonic begins to be identifiable at the tenth averaging.

Figure 14 shows the results of the Hanning window operation test. The signal is a square wave filtered at 25 Hz, like that in Fig. 13a. The scale of the ordinate is 0.12 V/√Hz per inch in this test.

The amplitude spectrum of the signal obtained without using the window is shown in Fig. 14a, and that obtained with the window, in Fig. 14b. The results foreseen theoretically (Par. 2.4), i.e., a decrease in resolution and a contemporaneous, clearer defintion of the spectral lines, were verified experimentally, although the differences do not seem to be remarkable.

Figures 14c and 14e show the spectra of the same signal embedded in noise, with S/N = 0.65, after the first and the 30th averaging, without the use of the window.

The same processing was repeated using the window, and the results are shown in Figs. 14d and 14f.

The results do not permit a conclusion to be drawn as to whether or not the window is useful. /35

In Figs. 15, 16 and 17, the same tests are repeated, substituting for the square wave a rectangular pulse with fundamental frequency 5 Hz and duration equal to about 1 octave of period (T/8): the test results do not differ substantially from those obtained previously.

The system's processing time, defined as the sum of the data acquisition time and the time of calculation of one spectrum, including the time for calculation of the spectral averaging, is shown in Table 1. The data output times have been excluded, for all of the terminals provided for.

TABLE 1.

Number of Samples	Acquisition Time	Calculation Time
512	5.1 sec	2.4 sec
256	2.5 sec	1.1 sec

The acquisition time depends on the sampling period, in our case fixed at 10 msec; with the use of a fast analog-to-digital converter, this time can be reduced to a negligible fraction of the calculation time.

The calculation time indicates the limit of the system as regards the frequency response. In fact, for a system in real time, in which acquisition and processing occur contemporaneously, /36
if N is the number of samples and T the calculation time, the maximum sampling frequency is given by:

$$f_C = N/T$$

and the signal band is half of f_C. In our case, with N = 512, the sampling frequency is about 200 Hz, and the signal band can thus be 100 Hz.

To increase the speed of calculation, and therefore the signal band, the commercial Fourier analyzers are equipped with fast arithmetic units connected to a standard computer and designed with the specific purpose of carrying out the FFT; in this way, calculation times on the order of a few msec are obtained, as are signal bands of a few tens of kHz [13].

5. Conclusion /37

The main purpose of this work was to study the feasibility of a system of data acquisition and processing in real time, using a small computer. It is known that for some time very elaborate systems have been available on the market, as regards both the hardware and the software for digital analysis of data in real time; but because their cost is high, such systems are justifiable only in particular cases in which their use would be extended in time and where other, more economical systems (such as, e.g., analog) cannot be substituted.

The results obtained with the present work show that if a suitable computer and the standard laboratory instrumentation are available, it is possible to construct a system that carries out some particular operations with performance comparable to that of the digital Fourier analyzers. The path followed to attain this objective was the preparation of a library of programs of data acquisition and analysis that can be called by a main program which can change, according to necessity, and which serves as an operative system to control the whole apparatus.

Some possible developments of the present system are easy to point out.

As far as the hardware is concerned, it is clear that perceptible advantages could derive from the use of a fast analog-to-digital converter: there are instruments on the market suited to /38
this use with relatively low costs, and with high frequency of conversion (e.g., 50 kHz and greater). An instrument of this type could reduce the data acquisition time to negligible values, and thus raise the signal bands analyzable in real time; in the case of stationary signals, the analysis could be extended to the maximum frequencies compatible with the speed of the converter, by separating the two phases of acquisition and processing of the signals.

Improvements in the software are possible, for example, extending the library of programs that can be called by the main program. For example, it can be useful to calculate the autocorrelation function of a signal,making use of the autocorrelation definition itself, and then to calculate the power spectrum by applying the Wiener-Khinchin theorem.

It is possible to consider processing two signals together, by means of the cross correlation functions or the crossed spectral density.

In every case, further developments in the analysis programs are linked to the type of application foreseen; thus it does not appear convenient to consider a system of general use based exclusively on a program library without contemporaneous development of special hardware components to be connected to the computer, in order that the system will be fast and easy to use.

The fact was stressed above that the two phases of data acquisition and processing are successive; a modification is /39
possible to make them contemporaneous, in the sense that the calculation is carried out on a group of data acquired previously, while continuing the acquisition of new data, by means of the machine's "interrupt" system. It is possible to modify the programs in this manner relatively easily; it should be noted that in the case of nonstationary signals, it is indispensable to analyze the data in real time, since it is incorrect to neglect components of the data sequence.

A further possibility that can be put into practice with little effort is to make the system adaptable to the spectral analysis of data acquired previously (e.g., data coming from satellites), and recorded on magnetic tape. It would thus be possible to analyze a large amount of data in a short time, without making use of rough systems of calculation.

In conclusion, it can be observed that a large number of problems can be confronted and resolved by a relatively economical digital system. The ever greater diffusion of small processers makes it ever more pressing to study systems with the purpose of carrying out signal analysis in real time, even if analog systems remain indispensable, especially in the analysis of high-frequency signals; they are often convenient due to their simplicity and their low cost. The analysis of each individual problem must suggest, on the basis of technical and economic considerations, the most suitable method of resolving it, and the instrumentation /40 best adapted for reaching the proposed objective.

Acknowledgements /41

I thank Eng. S. Cantarano and Eng. G.V. Pallotino for their frequent useful discussions of many problems and for their constant encouragement.

Appendices /44

Appendix A specifies the calls from BASIC to the programs that make up the Fourier analyzer library.

Appendix B shows the list for the main program of system control (MAIN), in BASIC language, and the list in ASSEMBLER of the FFT calculation subroutine. The flow charts of these two programs are in Figs. 8 and 9, respectively.

Appendix A: Program Library /45

1. FFT - Fast Fourier transform

```
CALL (50, N, L, K, B, D(1,1), E(1,1), W(1), S)
```

N - number of samples
L - $\log_2 N$
K = 0 - direct transform; K = 1 - inverse transform
B = 12 - length in bits of the words of vector W
D(1,1) - data matrix; at the beginning, contains the data acquired from the outside; at the end, the real component of the Fourier coefficients.
E(1,1) - data matrix; at the beginning, cleared; at the end, contains the imaginary component of the Fourier coefficients.
The dimensions of D and E, specified in the main program are:
DIM D(a,b), E(a,b)
with a*b = N/2.
W(1) - vector containing the values of the function $\cos(2\pi n/N)$ $|0 \leq n < N/4|$, multiplied by 2^B (B = 12) and transformed in integrals.
DIM W(a) with a = N/8.
S - Scale factor; must be cleared at input.

2. AVERAGE - Complex spectrum averaging

```
CALL (51, N, C(1,1), A(1,1), R(1,1), J, S1)
```

N - number of samples
C(1,1) - matrix of the spectral averages
DIM C(a,b) with a*b = N/4
A(1,1) - matrix of single spectrum
DIM A(a,b) with a*b = N/2
R(1,1) - matrix of inverse indices
DIM R(a,b) with a*b = N/2
J - number of calculation cycles
S1 - scale factor

3. ZERO - Matrix zero setting

```
CALL (52, N, A(1,1))

  N - matrix length
  A(1,1) - matrix to be cleared
           DIM A(a,b) with a*b = N/2.
```

4. WINDOW - Hanning window

```
CALL (53, N, D(1,1), W(1))
```

The parameters have the same significance as in the FFT call.

5. REVERSE - Calculation of inverse indices

```
CALL (54, N, R(1,1)
```

6. SAMPLE - Data acquisition

```
CALL (55, N, D(1,1)
```

The parameters have the same significance as in the FFT call.

7. POWER - Calculation of power spectrum

```
CALL (56, N, D(1,1), E(1,1), A(1,1))
```

The parameters have the same significance as in the FFT and AVERAGE calls.

```
10  DIM V[128,2]
20  DIM A[64,4],C[64,2]
30  DIM D[64,4],E[64,4]
40  DIM W[64],R[64,4]
50  DIM P[8],Q[4],H[4],L[4]
60  CALL (11,V[1,1],512)
70  CALL (14,M1)
75  CALL (41,1)
80  CALL (40,X,Y)
85  CALL (43,X,Y,-3)
90  LET M6=1
91  LET M7=10
92  LET M8=30
100  REM
110  REM  INITTIALIZATION
150  PRINT "NO OF SAMPLES AND LOG N";
160  INPUT N,L1
180  LET F=50
190  PRINT  "AVERAGE NOS"
200  INPUT M
210  LET B=12
220  REM  CALC COSINES AND INVERSE INDICES
230  GOSUB 5000
240  CALL (54,N,R[1,1])
250  REM  ZERO SETTING
260  LET N2=N/2
270  LET S1=0
280  CALL (52,N2,A[1,1])
300  CALL (52,N2,C[1,1])
310  REMFFT-0; PS-1
320  LET N1=1
330  REMHANN-1
340  LET W1=0
400  REM
410  REM  ACQUISITION
420  FOR J=1 TO M
430  CALL (52,N,D[1,1])
435  CALL (52,N,A[1,1])
440  CALL (52,N,E[1,1])
450  CALL (55,N,D[1,1])
452  CALL (1,11,X2)
454  IF X2 >= 0 THEN 460
455  IF J#1 THEN 460
456  LET Z=1
457  LET Z2=0
458  GOSUB 8000
460  IF W1=0 THEN 500
470  CALL (53,N,D[1,1],W[1])
500  REM
510  REM CALCULATION
520  LET S2=0
530  LET K1=0
540  CALL (50,N,L1,K1,B,D[1,1],E[1,1],W[1],S2)
```

```
550  LET S3=S1-S2
560  LET N3=2*N
570  CALL (56,N3,D[1,1],E[1,1],A[1,1])
580  CALL (51,N,C[1,1],A[1,1],R[1,1],J,S3)
600  IF S3 >= 0 THEN 700
610  LET S1=S2
700  REM
710  REMPRINT OUT
720  CALL (1,15,X)
730  IF X >= 0 THEN 800
740  GOSUB 6000
800  REM
810  REM VIDEO OUT
820  CALL (1,14,X1)
830  IF X1 >= 0 THEN 900
840  GOSUB 7000
900  REM
910  REM PLOTTER
920  CALL (1,11,X3)
930  IF X3 >= 0 THEN 1020
932  IF J=M6 THEN 940
934  IF J=M7 THEN 940
936  IF J=M8 THEN 940
938  GOTO 1020
940  REM PLOT OUT
1000  LET Z=-1
1010  GOSUB 8000
1020  NEXT J
1130  CALL (1,0,X4)
1140  IF X4 >= 0 THEN 400
1150  GOTO 150
4000  REM SAMPLING
4010  LET B1=2↑(16-L1)
4020  LET A1=1
4030  LET F1=2*3.14159*F/100
4040  LET N4=N/4
4043  LET I8=0
4045  LET I9=1
4050  FOR I=1 TO N STEP 4
4060  LET F3=F1*(I-1)
4065  FOR J9=1 TO 4
4070  LET P[J9]=A1*SIN(F3)*B1
4075  LET F3=F3+F1
4080  NEXT J9
4085  IF I8=1 THEN 4094
4087  CALL (30,P[1],D[I9,1],4)
4090  LET I8=1
4092  GOTO 4110
4094  CALL (30,P[1],D[I9,1],4)
4096  LET I8=0
4098  LET I9=I9+1
4110  NEXT I
4120  RETURN
```

```
5000  REM  COSINES
5010  LET N4=N/4
5020  LET T=2*3.14159/N
5030  FOR I=1 TO N4 STEP 2
5040  LET T1=T*(I-1)
5050  LET T2=T1+T
5060  LET P[1]=COS(T1)*2*B
5070  LET P[2]=COS(T2)*2*B
5080  LET I3=(I+1)/2
5090  CALL (30,P[1],W[I3],2)
5100  NEXT I
5110  RETURN
6000  REM
6010  REM PRINT OUT
6020  PRINT "SCALE"; 51
6030  LET N8=N/8
6040  FOR I=1 TO N8
6080  CALL (31,C[I,1],L[1],4)
6090  FOR I1=1 TO 4
6100  PRINT L[I1];
6110  NEXT I1
6112  PRINT
6115  CALL (1,15,X)
6116  IF X >= 0 THEN 6130
6120  NEXT I
6130  RETURN
7000  REM VIDEO
7010  LET N8=N/8
7020  IF N1>0 THEN 7600
7030  GOTO 7600
7600  REM POWER SP.
7610  LET X6=0
7630  LET N6=512/N
7640  FOR I=1 TO N8
7645  CALL (31,C[I,1],P[1],4)
7650  FOR K6=1 TO 4
7655  LET X6=X6+N6
7660  LET Y6=P[K6]/100+10
7670  CALL (12,X6,Y6)
7680  NEXT K6
7686  NEXT I
7780  CALL (1,14,X1)
7790  IF X1 >= 0 THEN 7800
7795  GOTO 7780
7800  CALL (14,M2)
7810  CALL (15,M1,M2)
7820  RETURN
8000  REM PLOT
8010  LET X=Y=0
8020  FOR I=1 TO 3
8030  LET Y=Y-.5
8040  CALL (43,0,Y,2)
8050  CALL (43,.1,Y,2)
8060  CALL (43,0,Y,2)
8070  NEXT I
```

```
8080  CALL (43,0,-2,-2)
8090  IF Z=-1 THEN 8110
8100  CALL (43,0,1,-3)
8110  LET X=Y=0
8120  FOR I=1 TO 5
8130  LET X=X+.5
8140  CALL (43,X,0,2)
8150  CALL (43,X,.1,2)
8160  CALL (43,X,0,2)
8170  NEXT I
8180  CALL (43,0,0,3)
8190  LET N8=N/8
8200  LET X=Y=0
8210  LET D5=(512/N)*1.00000E-02
8212  IF Z=-1 THEN 8220
8214  LET D5=2.5*D5
8220  FOR I=1 TO N8
8230  IF Z=-1 THEN 8290
8250  LET C5=350
8260  LET C6=8
8270  CALL (31,D[I,1],P[1],8)
8280  GOTO 8390
8290  LET C5=7000
8300  LET C6=4
8310  CALL (31,C[I,1],P[1],4)
8320  GOTO 8390
8390  FOR I1=1 TO C6
8400  LET Y=P[I1]/C5
8410  CALL (43,X,Y,2)
8420  LET X=X+D5
8425  IF X>2.5 THEN 8500
8430  NEXT I1
8440  NEXT I
8500  IF Z=-1 THEN 8560
8510  LET X=3
8520  LET Y=.75
8530  LET R9=8800
8540  LET R8=N
8550  GOTO 8600
8560  LET X=3
8570  LET Y=1.75
8580  LET R9=8810
8590  LET R8=J
8600  CALL (45,X,Y,.125,0,R9)
8610  LET Y=Y-.25
8620  LET R9=R9+1
8630  CALL (45,X,Y,.1,0,R9)
8640  LET X=X+.4
8650  CALL (44,X,Y,.1,0,R8)
8660  LET X=2.2
8670  LET Y=-.15
8680  IF Z=-1 THEN 8710
```

```
8690  LET R8=1
8695  LET X=2.5
8700  GOTO 8720
8710  LET R8=F
8720  CALL (44,X,Y,.1,0,R8)
8730  LET X=X+.3
8740  LET R9=R9+1
8750  CALL (45,X,Y,.1,0,R9)
8800  REM DATA RECORD
8801  REM N=
8802  REMSEC
8810  REMPOWER SPECTRUM
8811  REMAV=
8812  REMHZ
8900  IF Z=-1 THEN 8920
8910  CALL (43,0,-1,-3)
8920  IF M=1 THEN 8972
8925  IF Z2=3 THEN 8960
8930  CALL (43,0,-.5,-3)
8940  LET Z2=Z2+1
8950  GOTO 8980
8960  CALL (43,7,9.5,-3)
8970  LET Z2=0
8972  IF M#1 THEN 8980
8973  IF Z2=0 THEN 8977
8974  CALL (43,7,4.5,-3)
8975  LET Z2=0
8976  GOTO 8980
8977  CALL (43,0,-.5,-3)
8978  LET Z2=Z2+1
8980  RETURN
9999  END
```

```
      ORG 17150B
N     NOP
L1    NOP
K     NOP
IB    NOP
IRAD  NOP            REAL BUFFER
IMAD  NOP            IMAG BUFFER
IWAD  NOP            W    BUFFER
SCALA NOP
FFT   NOP
      JSB ENTR,I
      DEF N
      CLA
      STA INDOV      CLEAR OVF INDEX
      STA SCA         AND SCALE FACTOR
      CLA,INA
      STA N2         N2=1
      DLD N,I
      JSB IFIX
      NOP
      STB NNN
      STB N1         N1=N
      BLS
      STB N2         N2=2*N
      DLD L1,I
      JSB IFIX
      NOP
      CMB,INB
      STB L          FOR L=1 TOL1
LOOP1 DLD IB,I
      JSB IFIX
      NOP
      STB IBIT       IBIT=IB (12)
      CLA
      STA BITIN      CLEAR BITIN
      LDA INDOV
      SZA,RSS        OVF INDEX ZERO?
      JMP OK1        YES
      CLA            NO
      STA INDOV
      ISZ BITIN      BITIN=1
      ISZ SCA        SCA =SCA+1
OK1   LDA N2
      STA N1         N1=N2=2↑(L-1)
      ALS
      STA N2         N2=2*N1
      LDA N1
      ARS
      STA N1         N1=N1/2=N*2↑(-L)
      LDA N2
      ARS
      STA N2         N2=N2/2
      CMA,INA
      STA N3         N3=-N2
      LDA N1
```

```
        CMA,INA
        STA M          FOR M=1 TO M1
LOOP2   LDA M3
        ADA N2
        STA M3         M3=M3+N2
        LDA M1
        CMA,INA
        STA I3         I3=-M1
        CLA
        STA I          I=0
        LDA N1
        CMA,INA
        STA ICONT      FOR I=1 TO N1
LOOP3   LDA M3
        ADA I
        STA I1         I1=M3+I
        ISZ I
        LDA I3
        ADA M1
        STA I3         I3=I3+M1
        LDB I1
        ADB IRAD
        LDA 1,I
        ADB N1
        ADA 1,I
        JSB CONFR
        STA IS1        IS1=REAL(I1)+REAL(I1+N1)
        LDB I1
        ADB IRAD
        ADB N1
        LDA 1,I
        CMA,INA
        LDB I1
        ADB IRAD
        ADA 1,I
        JSB CONFR
        STA ID1        ID1=REAL(I1)-REAL(I1+N1)
        LDB I1
        ADB IMAD
        LDA 1,I
        ADB N1
        ADA 1,I
        JSB CONFR
        STA IS2        IS2=IMAG(I1)+IMAG(I1+N1)
        LDB I1
        ADB IMAD
        ADB N1
        LDA 1,I
        CMA,INA
        LDB I1
        ADB IMAD
        ADA 1,I
        JSB CONFR
        STA ID2        ID2=IMAG(I1)-IMAG(I1+N1)
        LDA I3
        SZA,RSS
```

```
         JMP P10
         LDA NNNN
         ARS,ARS
         CMA,INA
         ADA I3
         SZA,RSS
         JMP P30       I3=N/4
         SSA
         JMP P40       I3<N/4
         LDB I3        I3>N/4
         CMB,INB
         LDA NNNN
         ARS
         ADB 0
         STB I4        I4=N/2-I3
         ADB IWAD
         LDA 1,I
         CMA,INA
         STA IW1       IW1=-IW(I4)
         JMP P60
P40      LDB I3
         STB I4        I4=I3
         ADB IWAD
         LDA 1,I
         STA IW1       IW1=IW(I4)
P60      LDB I4
         CMB,INB
         LDA NNNN
         ARS,ARS
         ADB 0
         ADB IWAD
         LDA 1,I
         STA IW2       IW2=IW(N4-I4)
         DLD K,I
         JSB IFIX
         NOP
         STB KKK
         SZB,RSS       IF K=0 THEN DFT
         JMP P70
         CMA,INA
         STA IW2       IW2=-IW2
P70      LDA ID1
         MPY IW1
         DST M11
         LDA ID2
         MPY IW2
         CMB,CLE
         CMA,INA
         SEZ
         INB
         JSB ADD
         DEF M11
         JSB RED
         STA ID3       ID3=ID1*IW1-ID2*IW2
         LDA ID1
         MPY IW2
```

```
      DST M12
      LDA ID2
      MPY IW1
      JSB ADD
      DEF M12
      JSB RED
      STA ID4          ID4=ID1*IW2+ID2*IW1
      JMP P90
P30   LDB KKK
      SZB,RSS
      JMP P110
      LDA ID2          IFT
      STA ID3          ID3=ID2
      LDA ID1
      CMA,INA
      STA ID4          ID4=-ID1
      JMP P90
P110  LDA ID2          DFT
      CMA,INA
      STA ID3          ID3=-ID2
      LDA ID1
      STA ID4          ID4=ID1
      JMP P90
P10   LDA ID1
      LDB ID2
      STA ID3          ID3=ID1
      STB ID4          ID4=ID2
P90   LDB I1
      ADB IRAD
      LDA IS1
      STA 1,I          REAL(I1)=IS1
      ADB N1
      LDA ID3
      STA 1,I          REAL(I1+N1)=ID3
      LDB I1
      ADB IMAD
      LDA IS2
      STA 1,I          IMAG(I1)=IS2
      ADB N1
      LDA ID4
      STA 1,I          IMAG(I1+N1)=ID4
      ISZ ICONT
      JMP LOOP3        NEXTI
      ISZ M
      JMP LOOP2        NEXT M
      ISZ L
      JMP LOOP1        NEXT L
      LDA SCA
      JSB FLOAT,I
      DST SCALA,I
      JMP FFT,I
```

```
ADD     NOP            DOUBLE PRECISION ADDITION
        STB MM
        LDB ADD,I
        STB ADDR
        ISZ ADD
        CLE
        ADA ADDR,I
        ISZ ADDR
        LDB MM
        SEZ
        INB
        ADB ADDR,I
        JMP ADD,I
*
RED     NOP            DOUBLE TO SINGLE PRECISION
        DST MM          REDUCTION
        LDA IBIT
        CMA,INA
        INA
        STA BCONT
        DLD MM
LOOPS   CLE,ERB
        ERA
        ISZ BCONT
        JMP LOOPS
        CLE,INA
        SEZ
        INB
        RRR 1
        JMP RED,I
*
CONFR   NOP            OVERFLOW CONTROL
        LDB BITIN
        SZB
        ARS
        CLB
        RRL 3
        SZB,RSS
        JMP OK
        CPB D7
        JMP OK
        ISZ INDOV
OK      RRR 3
        JMP CONFR,I
```

```
*
*
M       NOP
M1      NOP
M2      NOP
M3      NOP
N1      NOP
N2      NOP
L       NOP
I       NOP
I1      NOP
I3      NOP
I4      NOP
ICONT   NOP
BCONT   NOP
ADDR    NOP
IS1     NOP
IS2     NOP
IS3     NOP
IS4     NOP
ID1     NOP
ID2     NOP
ID3     NOP
ID4     NOP
IW1     NOP
IW2     NOP
BITC    NOP
BITIN   NOP
SCA     NOP
IBIT    NOP
INDOV   NOP
D3      DEC 3
D7      DEC 7
M11     BSS 2
M12     BSS 2
MM      BSS 2
MMM     NOP
KKK     NOP
ENTR    EQU 61B
FLOAT   EQU 62B
IFIX    EQU 1364B
        END
```

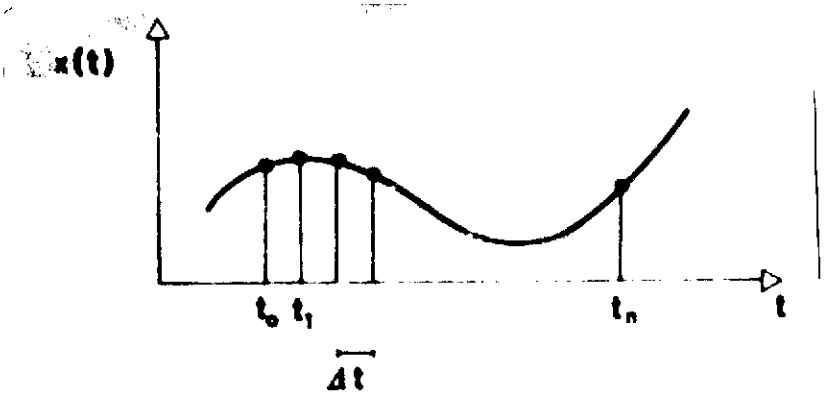


Fig. 2. Sampling of a continuous signal.

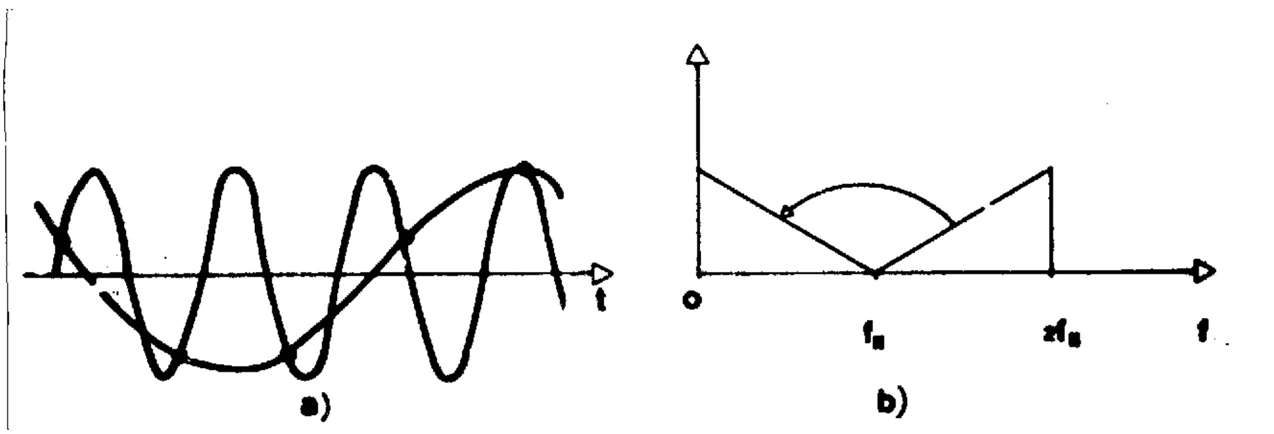


Fig. 3. Uncorrected sampling. Inversion of the spectrum (aliasing).

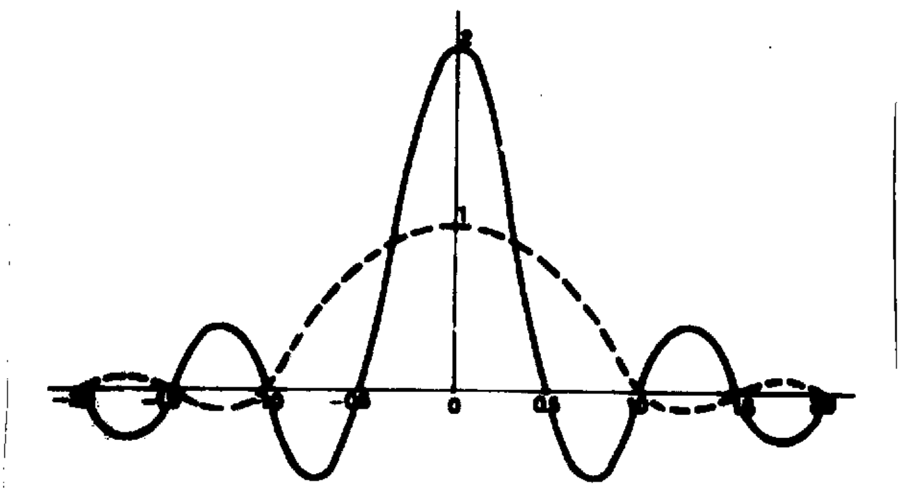


Fig. 4. Transform of the rectangular window (solid line) and the Hanning window.

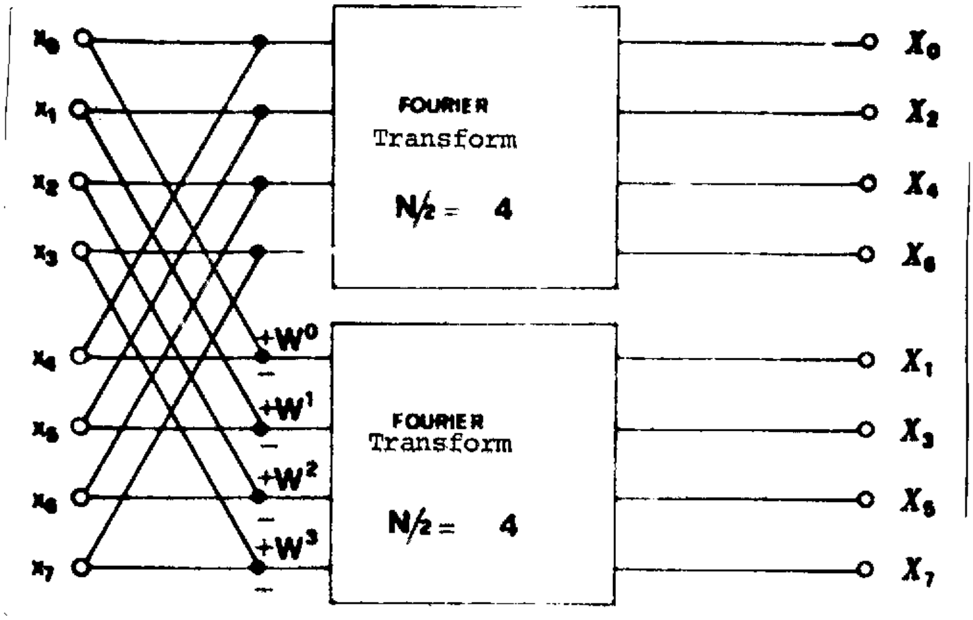


Fig. 5. Calculation of the Fourier transform of a series of N numbers with two transforms of N/2 numbers.

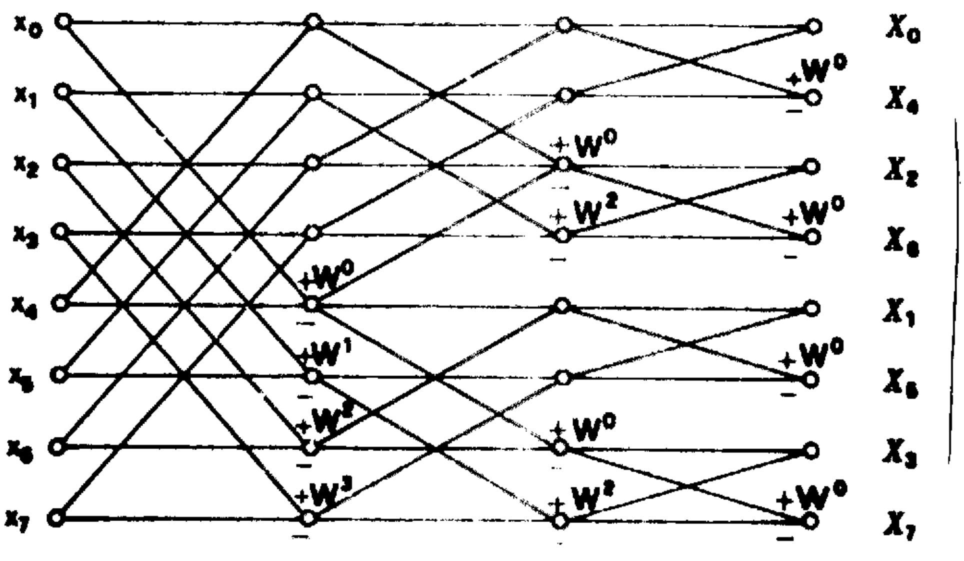


Fig. 6. Diagram of the calculation of the FFT for N = 8.

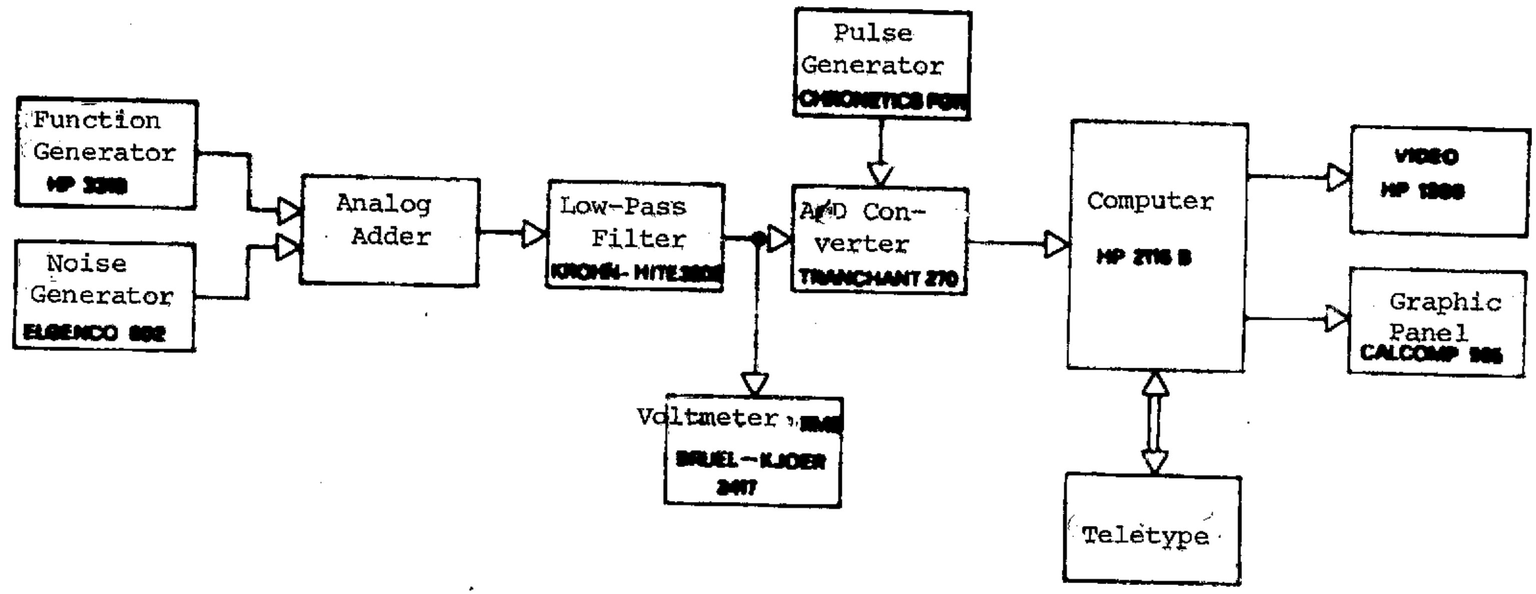


Fig. 7. Block diagram of the system for calculating the power spectrum of a continuous time function.

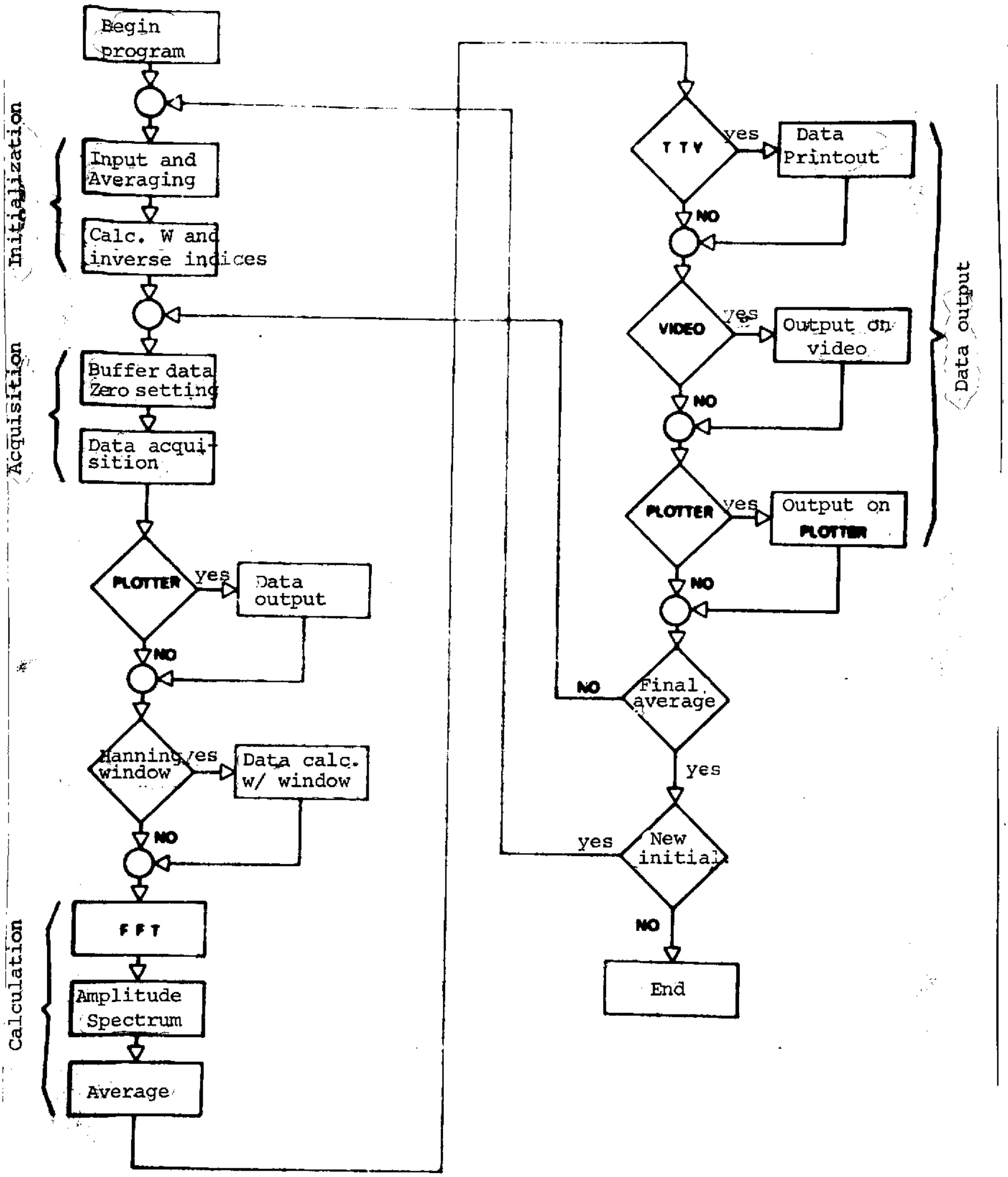


Fig. 8. Flow chart of main program for the calculation of the power spectrum.

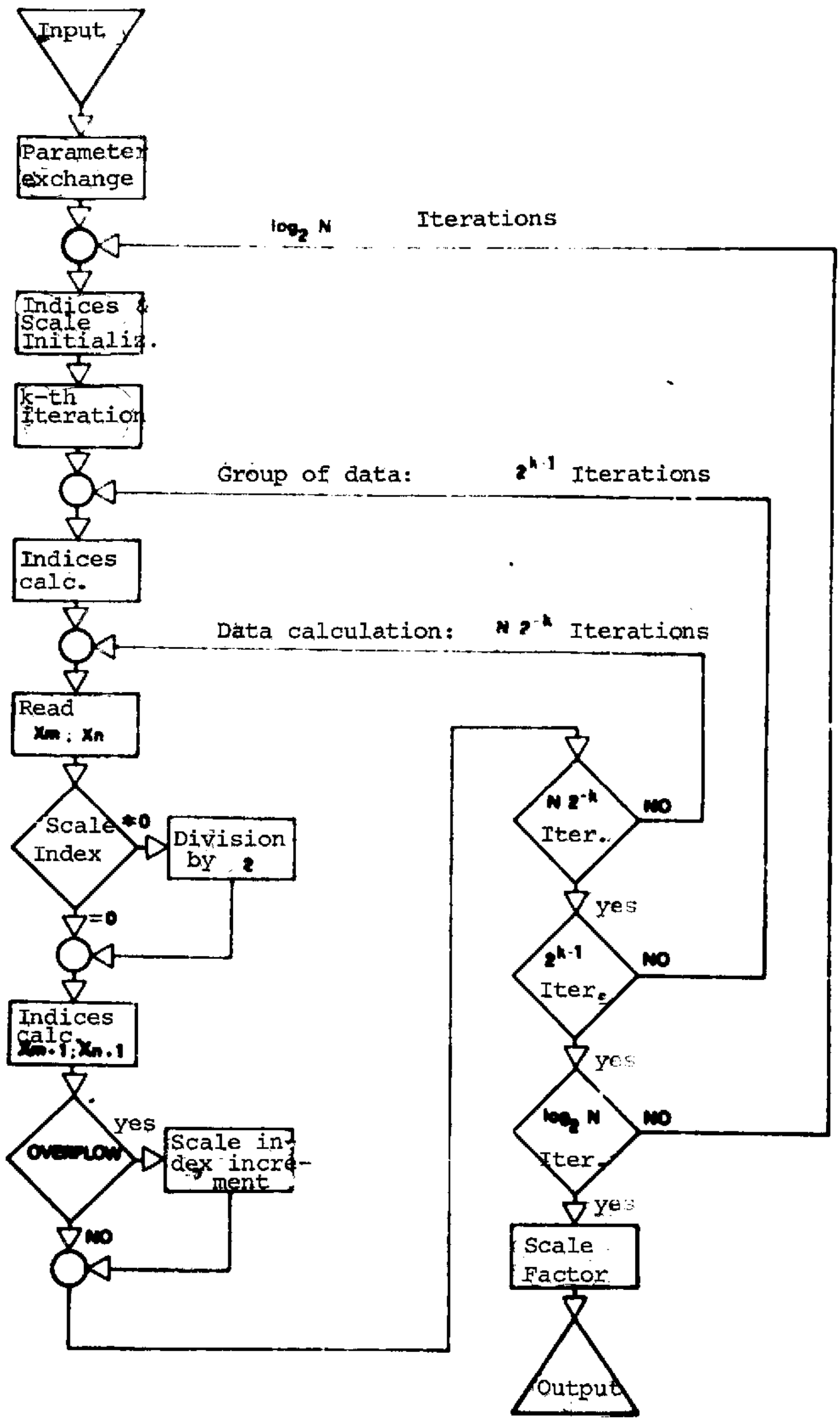


Fig. 9. Flow chart of the FFT subroutine.

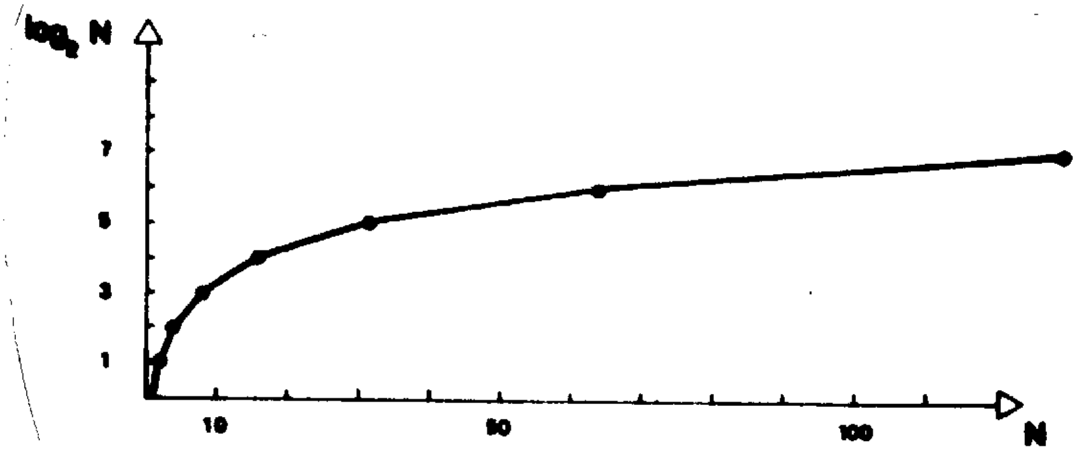


Fig. 10. Approximate binary logarithm of a number.

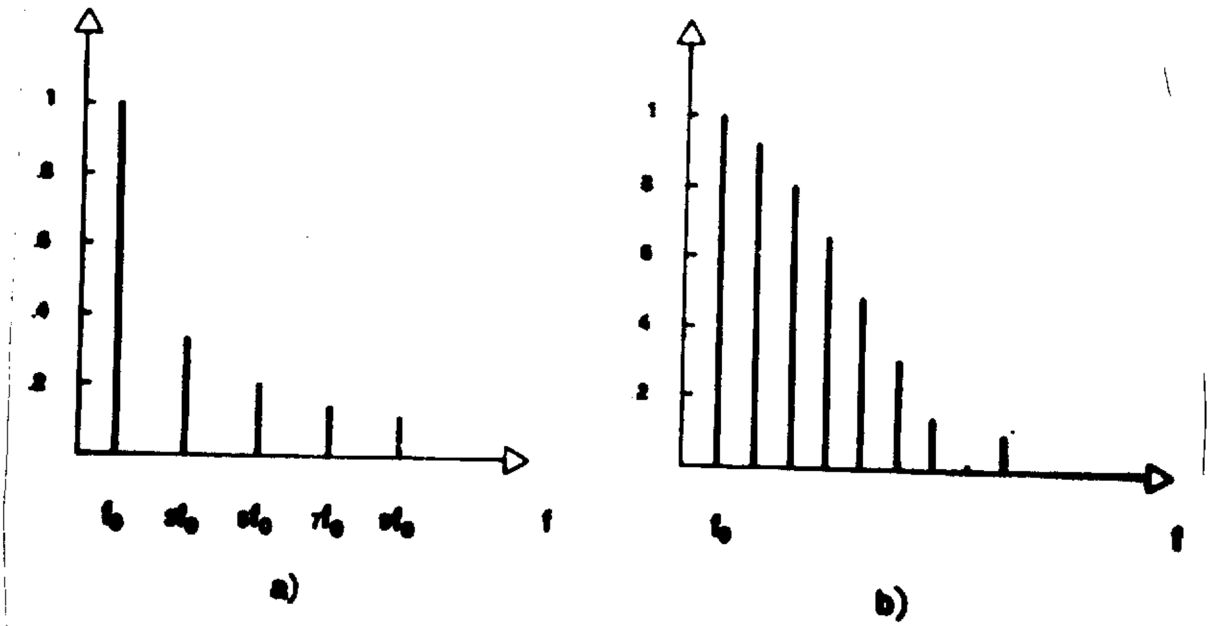
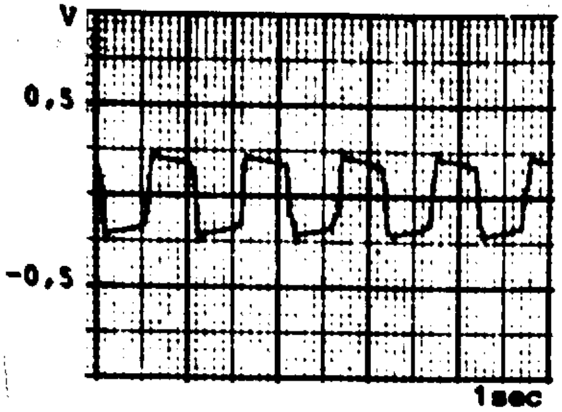
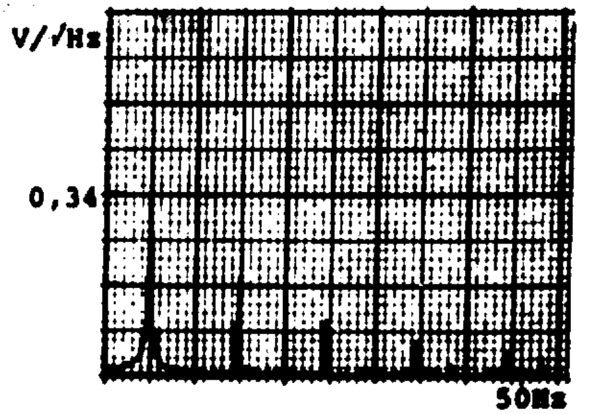


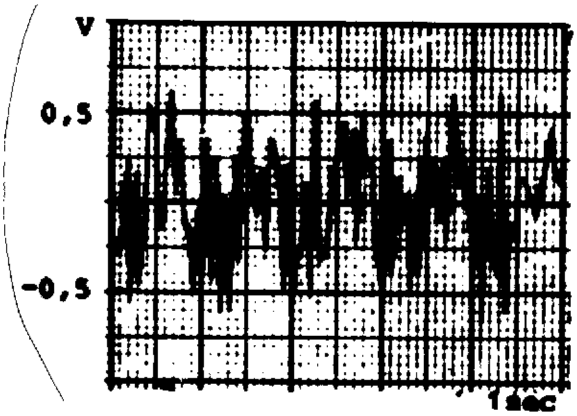
Fig. 11: a) Amplitude spectrum of a square wave.
b) Amplitude spectrum of a rectangular pulse.



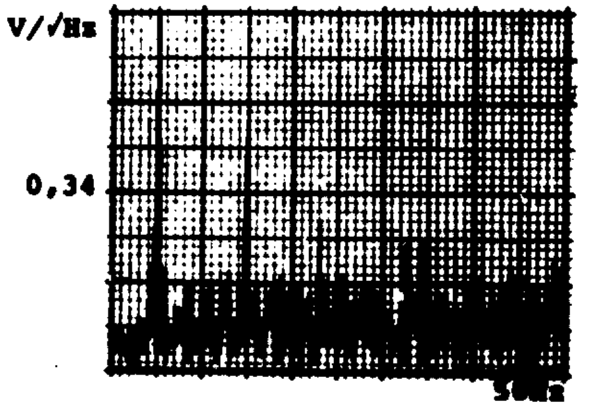
a) Square wave filtered at 50 Hz; record length: 5 sec; 512 samples per record; Nyquist frequency: 50 Hz.



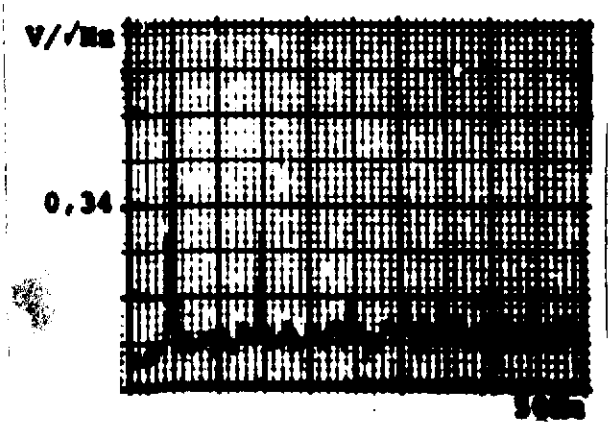
b) Amplitude spectrum of signal a; the 11th and 13th harmonics are seen to be inverted around the Nyquist frequency.



c) Square wave with noise (400 mV rms); signal/noise amplitude ratio: S/N = 0.6

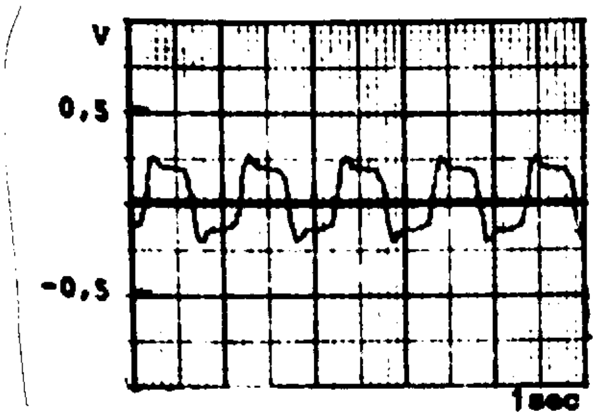


d) Single spectrum of signal c.

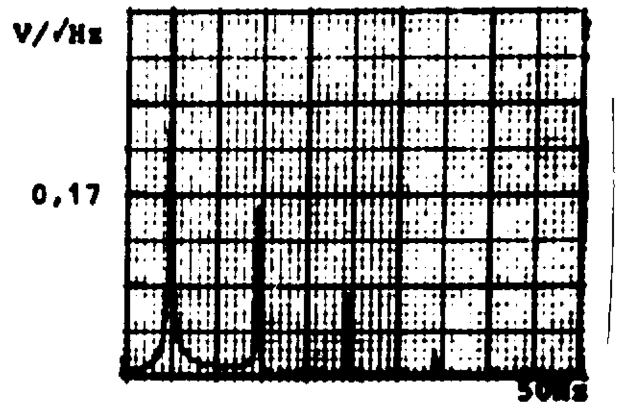


e) Spectrum of signal c after ten averagings; noise variance is decreased by a factor of 3.

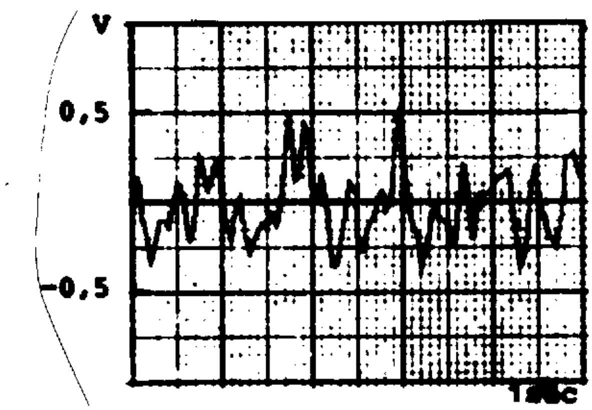
Fig. 12.



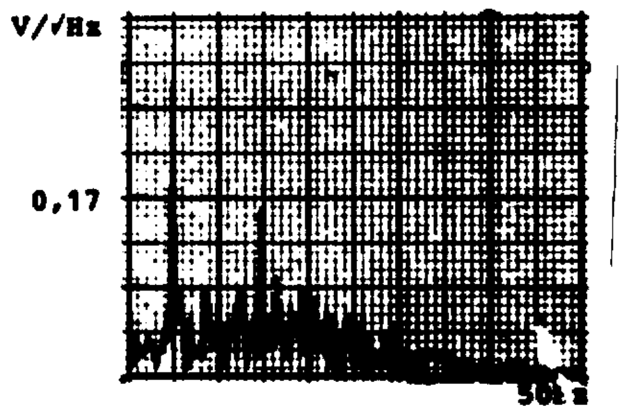
a) Square wave filtered at 25 Hz; record length: 5 sec; 512 samples per record; Nyquist frequency: 50 Hz.



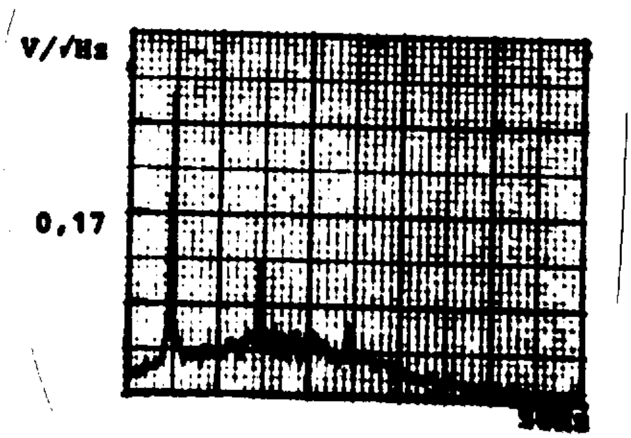
b) Amplitude spectrum of signal a; No harmonic frequency inversion greater than 50 Hz.



c) Square wave with noise (250 mV rms); S/N = 0.9

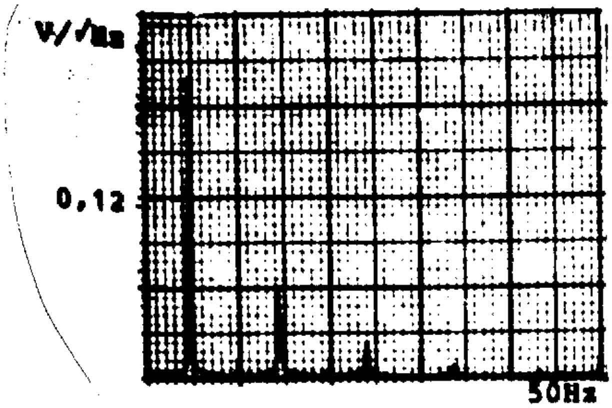


d) Single spectrum of signal c.

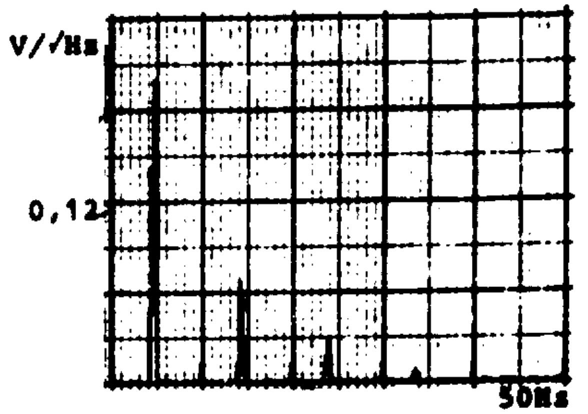


e) Spectrum of signal c after ten averagings.

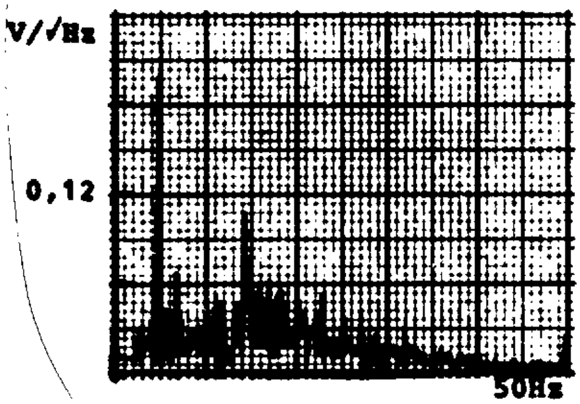
Fig. 13.



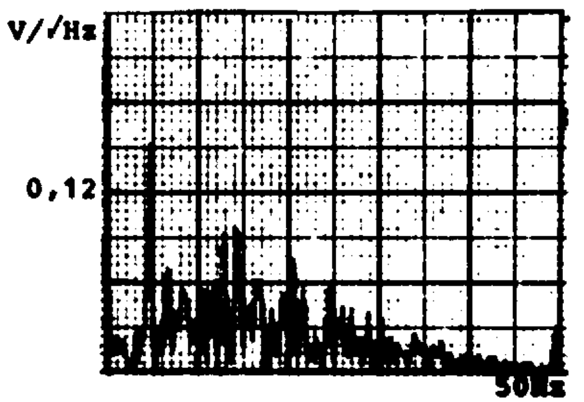
a) Spectrum of a square wave (100 mV rms); filter at 25 Hz; Nyquist frequency: 50 Hz; 512 samples per record.



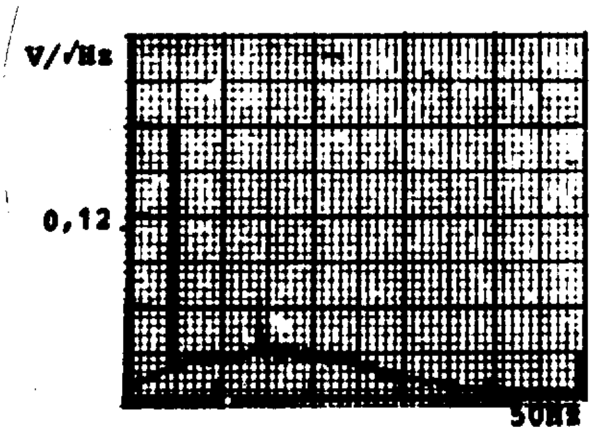
b) Spectrum of the same wave obtained by using a Hanning window; resolution decreases but leading and trailing edges are cancelled.



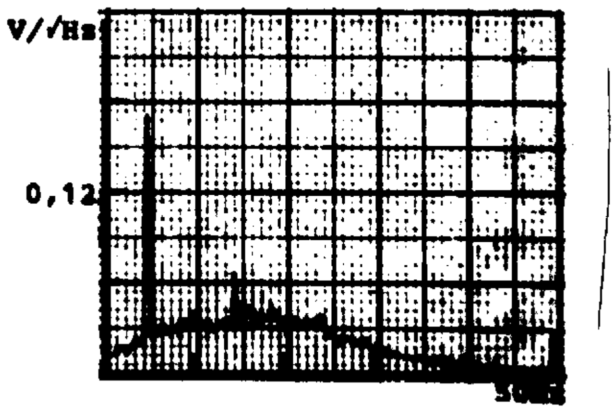
c) Single spectrum of signal a with S/N noise = 0.65; without window.



d) Single spectrum obtained by using the Hanning window.

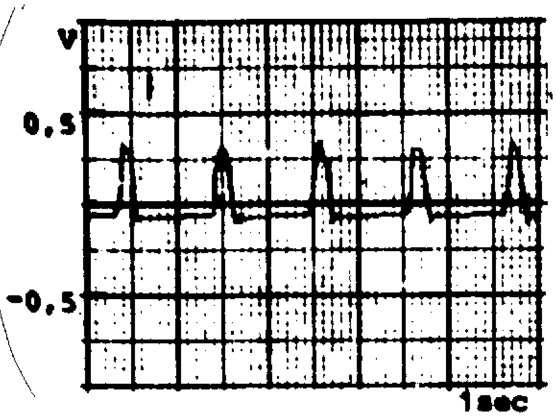


e) Spectrum obtained by averaging 30 spectra of type c; variance is diminished by about 5.

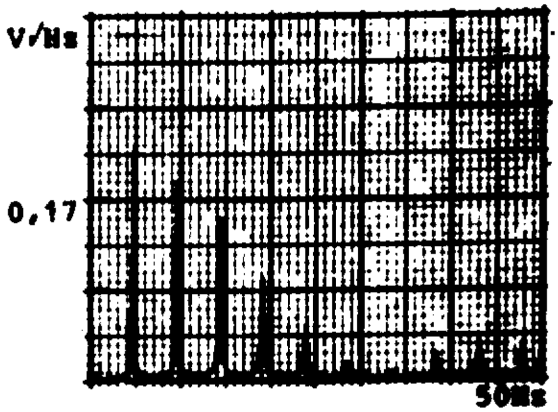


f) Spectrum obtained by averaging 30 spectra of type d.

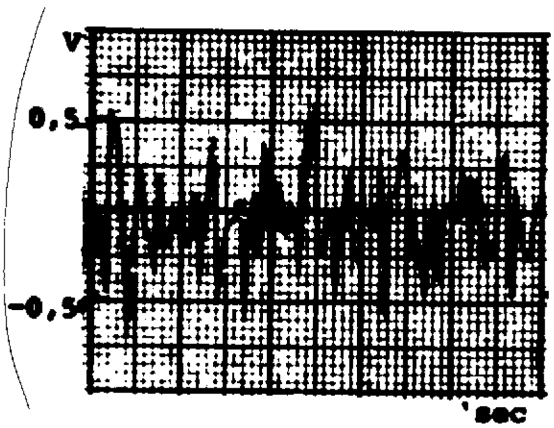
Fig. 14.



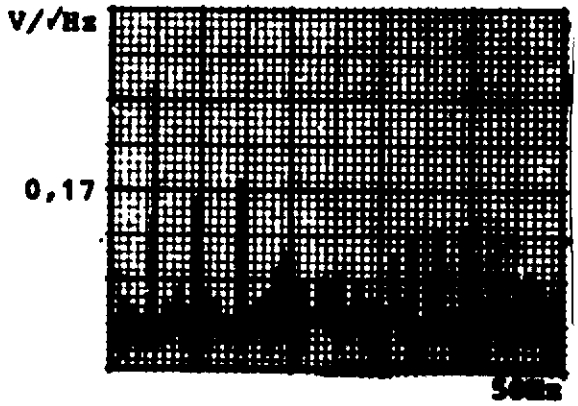
a) Pulse filtered at 50 Hz; record length: 5 sec; 512 samples per record, Nyquist frequency: 50 Hz.



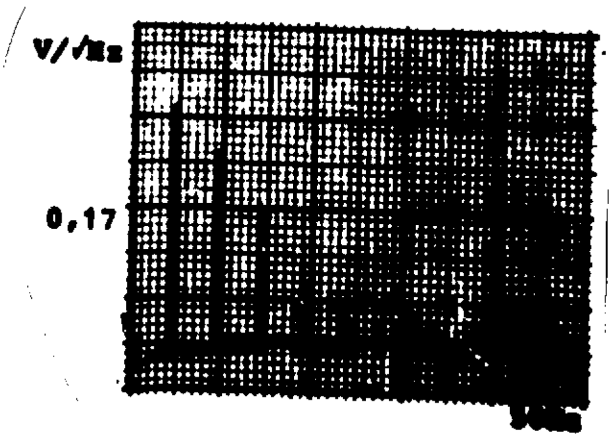
b) Amplitude spectrum of signal a.



c) Pulse with noise (400 mv rms); S/N ratio = 0.4 (in amplitude).

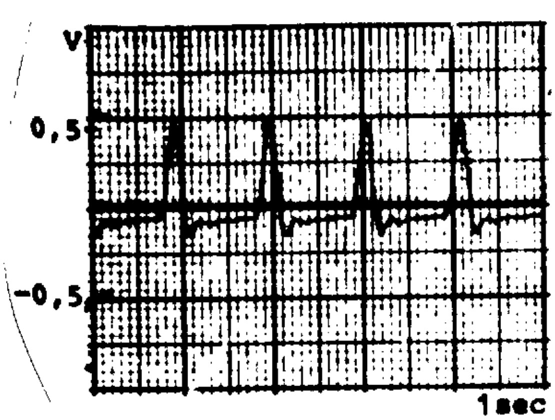


d) Single spectrum of signal c.

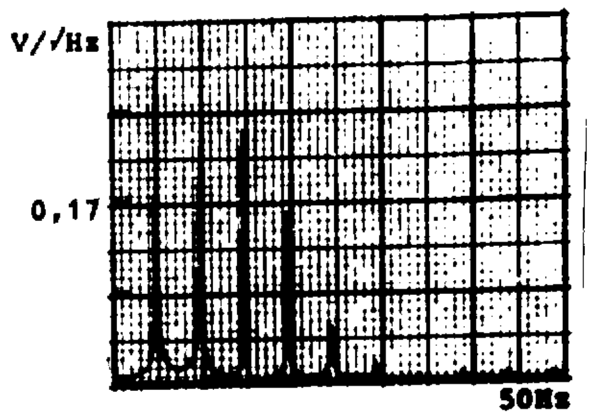


e) Spectrum of signal c after 30 averagings; the variance is diminished by a factor of 5.

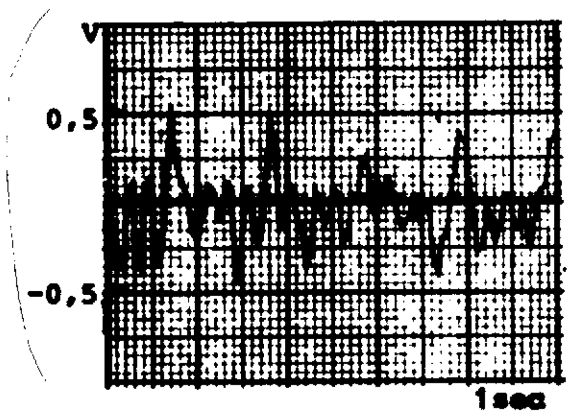
Fig. 15.



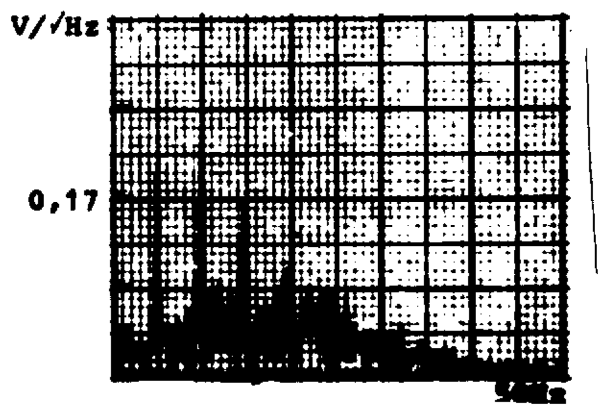
a) Pulse filtered at 25 Hz; record length: 5 sec; 512 samples per record, Nyquist frequency: 50 Hz.



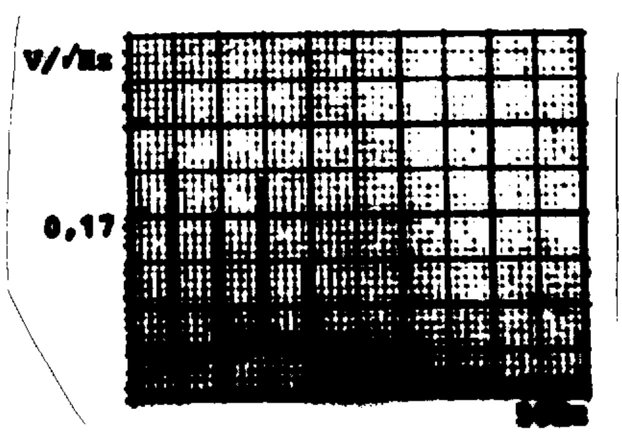
b) Amplitude spectrum of signal a.



c) Pulse with noise (250 mv rms); S/N ratio = 1.

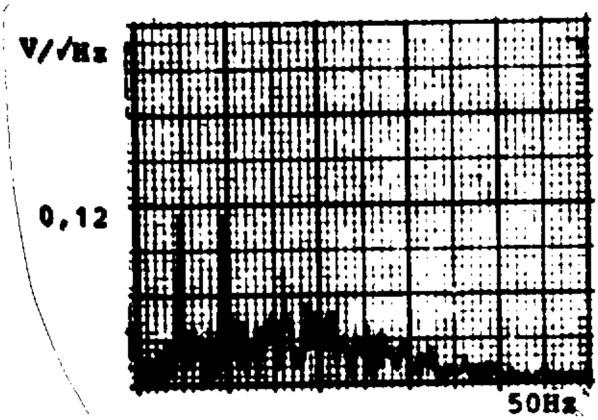


d) Single spectrum of signal c.

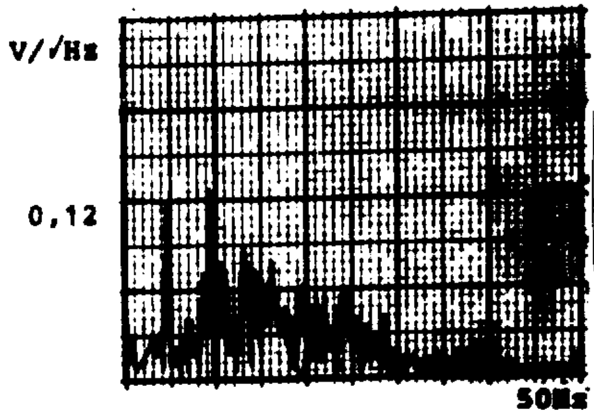


e) Spectrum of signal c averaged 30 times.

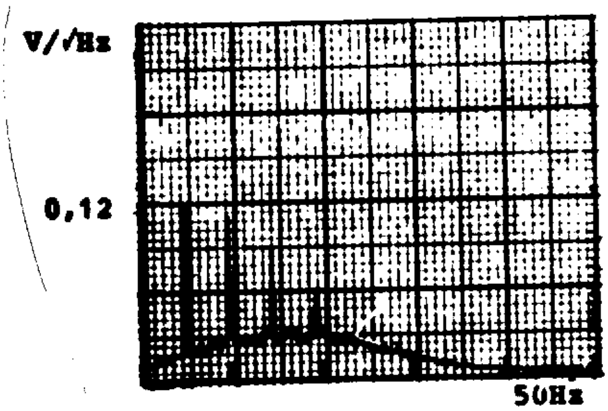
Fig. 16.



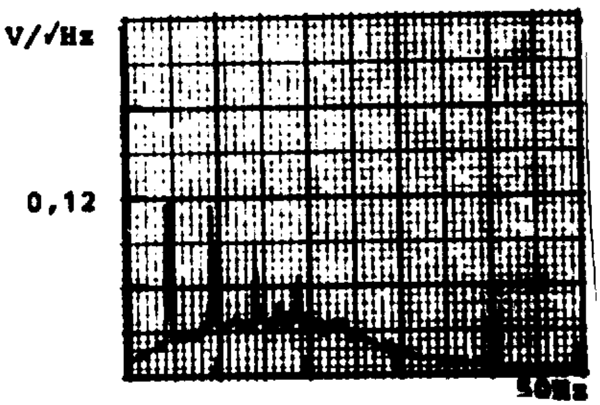
a) Amplitude spectrum of a pulse signal embedded in noise; S/N = = 0.6; filter at 25 Hz.



b) Spectrum of the same signal shaped with Hanning window.



c) Average of 30 spectra of type a.



d) Average of 30 spectra of type b.

Fig. 17.

REFERENCES

1. Fanton, J.L., "A computer-aided hospital system for cardiac catheterization procedures," HP Journal (Jan 1972).

2. Bendat, Piersol, Random data: Analysis and measurement procedures, J. Wiley, 1971.

3. Blackman, Tukey, The measurement of power spectra, Dover Publ. Inc., 1968.

4. Dotti, D., "Four different techniques to measure power and cross-power spectra," Alta frequenza 38/11 (Nov. 1969).

5. Roth, P.R., "Digital Fourier analysis," HP Journal (June 1970).

6. Kiss, A.Z., "A calibrated computer-based Fourier analysis," HP Journal (June 1970).

7. Schwartz, M., Information, transmission, modulation and noise, McGraw Hill Co., 1959, Chapter 4.

8. Cooley, Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comp. 19, 297-301 (1965).

9. Gold, Rader, Digital processing of signals, 1969 [sic].

10. IEEE Transactions on Audio and Electroacoustics: Special number devoted to the FFT AU-15/2 (June 1967). /43

11. Welch, P.D., "A fixed-point FFT error analysis," IEEE Trans. on Audio and Electroacoustics AU-17/2 (June 1969).

12. Mitchell, J.N., "Computer multiplication and division using binary logarithms," IRE Trans. on Electronic Computers (Aug. 1962).

13. Cline, S.G. and N.D. Marschke, "New capabilities in digital low-frequency spectrum analysis," HP Journal (June 1972).